



PUBLIC

SAP Data Services

Document Version: 4.2 Support Package 10 (14.2.10.0) – 2018-12-10

Reference Guide

Content

1	Introduction to SAP Data Services.	14
2	SAP information resources.	15
3	About this guide.	16
4	Who should read this guide.	17
5	Objects.	18
5.1	Object classes.	18
	Reusable objects.	18
	Single-use objects.	19
5.2	Object options, properties, and attributes.	20
5.3	Descriptions of objects.	20
	Annotation.	23
	Batch Job.	24
	Catch.	35
	COBOL copybook file format.	39
	Conditional.	49
	Data flow.	51
	Datastore.	56
	Document.	128
	DTD.	129
	Excel workbook format.	139
	File format.	147
	File location object.	171
	Function.	193
	JSON.	194
	Log.	196
	Message function.	200
	Nested Schemas template.	200
	Outbound message.	201
	Project.	202
	Query transform.	202
	Real-time job.	203
	Script.	207
	Source.	207
	Table.	221

	Target.	223
	Target Writer migrated from Data Quality.	274
	Template table.	274
	Transform.	277
	Try.	278
	While loop.	279
	Work flow.	279
	XML file.	281
	XML message.	284
	XML schema.	286
6	Smart editor.	299
6.1	Accessing the smart editor.	299
6.2	Smart editor toolbar.	300
6.3	Editor Library pane.	300
	Tabs.	300
	Find option.	301
6.4	Editor pane.	301
	Syntax coloring.	301
	Selection list and tool tips.	301
	Using the selection list and a tool tip.	301
	Right-click menu and toolbar.	302
6.5	Validating.	303
6.6	Browsing for a function.	303
6.7	Searching for a function.	304
7	Data Types.	305
7.1	Descriptions of data types.	305
	date.	305
	datetime.	307
	decimal.	308
	double.	308
	int (integer).	309
	interval.	309
	Limitations for long and blob.	309
	numeric.	313
	real.	313
	time.	314
	timestamp.	315
	varchar.	316
7.2	Data type conversion.	317
	Date arithmetic.	317

	Conversion to or from internal data types.	318
	Conversion of data types within expressions.	344
	Conversion among number data types.	344
	Conversion between explicit data types.	346
	Conversion between native data types.	347
8	Transforms.	351
8.1	Transform reference.	351
8.2	Dynamic transform settings.	354
8.3	Data Services Embedded Help.	356
8.4	Data Integrator transforms.	357
	Data_Transfer	358
	Date_Generation	372
	Effective_Date	375
	Hierarchy_Flattening	380
	History_Preserving	392
	Key_Generation	401
	Map_CDC_Operation	405
	Pivot (Columns to Rows)	413
	Reverse Pivot (Rows to Columns)	418
	Table_Comparison.	422
	XML_Pipeline	433
8.5	Data Quality transforms.	437
	Blueprints and other content objects for download.	438
	About Data Quality fields.	445
	About data quality statistics.	448
	Associate	454
	Country ID.	472
	Data Cleanse.	474
	DSF2® Walk Sequencer.	514
	Geocoder.	524
	Global Address Cleanse.	552
	Global Suggestion List	640
	Match.	651
	USA Regulatory Address Cleanse.	710
	Address Cleanse reference.	765
	Data Cleanse reference.	823
8.6	Platform transforms.	833
	Case.	833
	Data Mask.	836
	DQM Microservices.	925
	Map_Operation	938

	Merge	942
	Query	943
	Row_Generation	970
	SQL	972
	User-Defined	975
	Validation	983
	XML_Map	994
8.7	Text Data Processing transforms.	1013
	Entity Extraction transform.	1014
9	Functions and Procedures.	1035
9.1	About functions.	1035
	Functions compared with transforms.	1035
	Operation of a function.	1035
	Arithmetic in date functions.	1036
	Including functions in expressions.	1036
9.2	Built-in functions.	1038
	Database and application functions.	1038
9.3	Descriptions of built-in functions.	1039
	abs.	1047
	add_months.	1048
	ascii.	1049
	avg.	1050
	base64_decode.	1051
	base64_encode.	1051
	before_image.	1052
	begin_delta_load.	1053
	begin_initial_load.	1054
	cast.	1054
	chr.	1056
	ceil.	1057
	concat_date_time	1058
	copy_from_remote_system.	1058
	copy_to_remote_system.	1060
	count	1061
	count_distinct.	1062
	current_configuration	1063
	current_system_configuration.	1064
	dataflow_name	1064
	datastore_field_value.	1065
	date_diff.	1065
	date_part	1067

day_in_month.	1068
day_in_week.	1069
day_in_year.	1070
db_type.	1071
db_version.	1072
db_database_name.	1073
db_owner.	1074
decode.	1075
decrypt_aes.	1077
decrypt_aes_ext.	1078
double_metaphone.	1079
encrypt_aes.	1080
encrypt_aes_ext.	1081
error_context.	1082
error_message.	1083
error_number.	1084
error_timestamp.	1084
exec.	1085
extract_from_json.	1089
extract_from_xml.	1090
file_copy.	1091
file_delete.	1093
file_exists.	1094
file_move.	1095
fiscal_day.	1097
floor.	1098
gbq2file.	1099
gen_row_num_by_group.	1100
gen_row_num.	1102
gen_uuid.	1103
get_domain_description.	1103
get_env.	1104
get_error_filename.	1105
get_file_attribute.	1106
get_monitor_filename.	1107
get_trace_filename.	1107
greatest.	1108
host_name.	1110
ifthenelse.	1110
index.	1111
init_cap.	1112

interval_to_char.	1114
is_group_changed.	1115
is_set_env.	1116
is_valid_date.	1117
is_valid_datetime.	1118
is_valid_decimal.	1119
is_valid_double.	1120
is_valid_int.	1121
is_valid_real.	1122
is_valid_time.	1123
isempty.	1124
isweekend.	1125
job_name.	1126
Job_Run_ID.	1126
julian.	1127
julian_to_date.	1128
key_generation.	1128
last_date.	1129
least.	1130
length.	1131
literal.	1132
ln.	1134
load_from_gcs_to_gbq.	1135
load_from_s3_to_redshift.	1136
load_to_xml.	1139
local_to_utc.	1140
log.	1141
long_to_varchar.	1142
lookup.	1142
lookup_ext.	1146
lookup_seq.	1152
lower.	1155
lpad.	1156
lpad_ext.	1157
ltrim.	1158
ltrim_blanks.	1160
ltrim_blanks_ext.	1160
mail_to.	1161
match_pattern.	1164
match_regex.	1166
match_simple.	1171

max.	1173
min.	1174
mod.	1174
month.	1175
num_to_interval.	1176
nvl.	1177
power.	1178
previous_row_value.	1179
print.	1180
pushdown_sql.	1181
quarter.	1183
raise_exception.	1184
raise_exception_ext.	1185
rand.	1185
rand_ext.	1186
regex_replace.	1187
replace_substr.	1188
replace_substr_ext.	1189
repository_name.	1191
restore_repserver_cdb_backlogged_transactions.	1191
round.	1192
rpadd.	1193
rpadd_ext.	1194
rtrim.	1195
rtrim_blanks.	1196
rtrim_blanks_ext.	1197
sap_openhub_processchain_execute.	1198
sap_openhub_set_read_status.	1201
search_replace.	1202
set_cdc_checkpoint.	1207
set_env.	1207
sleep.	1208
soundex.	1209
sql.	1210
sqrt.	1212
smtp_to.	1212
string_to_number.	1215
substr.	1215
sum.	1217
sysdate.	1218
system_user_name.	1218

	systemtime.	1219
	table_attribute.	1220
	to_char.	1221
	to_date.	1224
	to_decimal.	1225
	to_decimal_ext.	1226
	to_WKT_point.	1227
	total_rows.	1228
	translate.	1229
	trunc.	1230
	truncate_table.	1231
	upper.	1232
	utc_to_local.	1233
	varchar_to_long.	1234
	wait_for_file.	1235
	week_in_month.	1236
	week_in_year.	1237
	WL_GetKeyValue.	1238
	word.	1239
	word_ext.	1240
	workflow_name.	1241
	year.	1241
9.4	Custom functions.	1242
	Creating a custom function.	1243
	Editing an existing function.	1244
	Replicating a custom function.	1245
	Deleting a custom function.	1245
9.5	About procedures.	1245
	Overview.	1245
	Requirements.	1246
	Creating stored procedures in a database.	1247
	Importing metadata for stored procedures.	1251
	Structure of a stored procedure.	1252
	Calling stored procedures.	1253
	Checking execution status.	1257
10	Scripting Language.	1259
10.1	Using the scripting language.	1259
10.2	Language syntax.	1259
	Syntax for statements in scripts.	1259
	Syntax for column and table references in expressions.	1260
	Strings.	1260

	Variables.	1262
	Variable interpolation.	1262
	Functions and stored procedures.	1263
	Operators.	1263
	NULL values.	1265
	Debugging and Validation.	1267
	Keywords.	1268
10.3	Sample scripts.	1270
	Square function.	1270
	RepeatString function.	1271
11	Metadata in Repository Tables and Views.	1272
11.1	Auditing metadata.	1272
	AL_AUDIT.	1272
	AL_AUDIT_INFO.	1273
11.2	Imported metadata.	1274
	AL_INDEX.	1274
	AL_PCOLUMN.	1274
	AL_PKEY.	1275
	ALVW_COLUMNATTR.	1275
	ALVW_COLUMNINFO.	1276
	ALVW_FKREL.	1277
	ALVW_MAPPING.	1278
	ALVW_TABLEATTR.	1284
	ALVW_TABLEINFO.	1284
11.3	Internal metadata.	1285
	AL_LANG.	1285
	AL_LANGXMLTEXT.	1286
	AL_ATTR.	1286
	AL_SETOPTIONS.	1287
	AL_USAGE.	1287
	ALVW_FUNCINFO.	1290
	ALVW_PARENT_CHILD.	1291
11.4	Operational metadata.	1292
	AL_HISTORY.	1292
	DQVW_AGR_RPTSTATS.	1293
	ALVW_FLOW_STAT.	1293
	DQVW_REPORTS_STAT.	1294
12	Data quality reports and statistics tables	1296
12.1	Repository tables common columns.	1297
12.2	Repository tables and related reports	1297

	Repository tables for USA and Global address cleanse.	1299
	Repository tables and reports for Data Cleanse.	1321
	Repository tables and reports for Geocoder.	1323
	Match repository statistics tables.	1325
12.3	Data quality statistics tables and supplemental content information.	1344
	Content information for data quality statistics tables.	1346
	CLEANSE_ADDRESS_RECORD_INFO_.	1352
	CLEANSE_CHANGE_INFO_.	1356
	CLEANSE_COMPONENT_INFO_.	1360
	CLEANSE_INFO_CODES_.	1366
	CLEANSE_STATISTICS_.	1370
	GEOCODE_INFO_CODE_.	1373
	GEOCODE_STATISTICS_.	1376
13	Locales and Multi-byte Functionality.	1378
13.1	Language packs.	1378
	Setting locales in the Designer.	1378
	Setting locales in the Locale Selector.	1379
	Setting locales in UNIX or Linux.	1379
	Impact of locale settings on Data Services components.	1380
13.2	Locale support.	1380
	Locale selection.	1382
	Code page support.	1384
	Guidelines for setting locales.	1386
13.3	Multi-byte support.	1391
	Multi-byte string functions.	1391
	Numeric data types: assigning constant values.	1392
	Byte Order Mark characters.	1393
	Round-trip conversion.	1393
	Column sizing.	1393
13.4	Limitations of multi-byte support.	1394
13.5	Definitions.	1394
13.6	Supported locales and encodings.	1397
14	Python.	1400
14.1	About Python.	1400
	Python module.	1400
	Supported transform.	1400
	Third-party Python libraries.	1401
	Processing mode.	1401
	Internal coding.	1401
	Unicode in Python.	1401

	Clean up new memory references.	1402
	Using Mapped_Name.	1402
	Using paths with substitution parameters and custom options.	1403
14.2	Creating an expression with the Python Expression editor.	1403
	Opening the Python Expression editor from the User-Defined transform.	1404
	Opening the Python Expression editor from the Match transform.	1404
	Writing Python code.	1405
	Validating syntax.	1406
	Fixing syntax.	1406
	Find and Replace.	1406
14.3	Built-in objects.	1407
14.4	Defined classes and methods.	1407
14.5	FIDataCollection class.	1408
	AddRecord.	1408
	DeleteRecord.	1410
	GetRecord.	1410
	Size.	1411
	Truncate.	1412
14.6	FIDataManager class.	1413
	DeleteDataRecord.	1413
	NewDataRecord.	1414
14.7	FIDataRecord class.	1416
	GetField.	1416
	SetField.	1417
14.8	FIProperties class.	1418
	GetProperty.	1418
14.9	FIPythonString class.	1419
	GetBuffer.	1420
	SetBuffer.	1420
14.10	Python examples.	1421
	Formatting data.	1422
	Splitting data.	1423
	Best Record.	1425
	Assigning source attributes.	1425
15	Hadoop overview.	1427
15.1	Hadoop in Data Services.	1429
15.2	Hadoop sources and targets.	1431
15.3	Prerequisites to Data Services configuration.	1432
15.4	Verify Linux setup with common commands	1433
	Configuring Hadoop for text data processing.	1434
15.5	Hadoop support for the Windows platform.	1435

	Setting up HDFS and Hive on Windows.	1436
15.6	Connect to HDFS.	1437
	Previewing HDFS file data.	1438
15.7	Connect to Hive.	1438
16	Reserved Words.	1440
16.1	About Reserved Words.	1440
17	Glossary.	1443

1 Introduction to SAP Data Services

SAP Data Services delivers a single enterprise-class solution for data integration, data quality, data profiling, and text data processing.





Businesses can use Data Services to integrate, transform, improve, and deliver trusted data to critical business processes. IT organizations can depend on Data Services for maximum operational efficiency to improve data quality and gain access to heterogeneous sources and applications. Data Services provides all of these features using:

- Single development user interface
- Metadata repository
- Data connectivity layer
- Runtime environment
- Management Console

2 SAP information resources

SAP supplies you with a global network of technology experts including customer support, education, and consulting to ensure maximum information management benefit to your business.

The following table lists the resources that are available to you as a user of SAP Data Services.

Resource	Content
Customer Support, Consulting, and Education services 	Information about SAP Business User Support programs, as well as links to technical articles, downloads, and online discussions.
Product documentation and tutorial	<p>SAP product documentation.</p> <p>The tutorial introduces core features, concepts and techniques to extract, transform, and load batch data from flat-file and relational database sources for use in a data warehouse.</p>
SAP Data Services on SAP Community 	Get online and timely information about SAP Data Services, including forums, tips and tricks, additional downloads, samples, and much more. All content is to and from the community, so feel free to join in and contact us if you have a submission.
Product Availability Matrix (PAM) 	Information about supported platforms for SAP Data Services with a search function to quickly find information related to your platform.
Blueprints 	Blueprints for you to download and modify to fit your needs. Each blueprint contains the necessary SAP Data Services project, jobs, data flows, file formats, sample data, template tables, and custom functions to run the data flows in your environment with only a few modifications.

3 About this guide

The *Data Services Reference Guide* provides detailed information about the objects, data types, transforms, and functions in the Designer.

For source-specific information, such as information pertaining to a particular back-office application, refer to the supplement for that application.

4 Who should read this guide

This and other SAP Data Services software documentation assume the following:

- You are a software developer, consultant, or database administrator working on data extraction, data warehousing, data integration, or data quality.
- You understand your source and target data systems, DBMS, legacy systems, business intelligence, and messaging concepts.
- You understand your organization's data needs.
- You are familiar with SQL (Structured Query Language).
- If you are interested in using this software to design real-time processing, you are familiar with:
 - DTD and XML Schema formats for XML files
 - Publishing Web Services (WSDL, HTTP/S and SOAP protocols, and so on.)
- You are familiar with the installation environments: Microsoft Windows or UNIX.

5 Objects

All "entities" you define, edit, or work with in Designer are called objects.

i Note

For information about source-specific objects, consult the reference section in the supplement document for that source.

Related Information

[Object classes \[page 18\]](#)

[Object options, properties, and attributes \[page 20\]](#)

[Descriptions of objects \[page 20\]](#)

5.1 Object classes

An object's class determines how you create and retrieve the object.

There are two classes of objects:

- Reusable objects
- Single-use objects

5.1.1 Reusable objects

You can reuse and replicate most objects defined in the software.

After you define and save a reusable object, SAP Data Services stores the definition in the repository. You can then reuse the definition as often as necessary by creating calls to the definition. You access reusable objects through the object library.

A reusable object has a single definition; all calls to the object refer to that definition. If you change the definition of the object in one place, and then save the object, the change is reflected to all other calls to the object.

A data flow, for example, is a reusable object. Multiple jobs, such as a weekly load job and a daily load job, can call the same data flow. If the data flow is changed, both jobs call the new version of the data flow.

When you drag and drop an object from the object library, you are creating a new reference (or `<call>`) to the existing object definition.

You can edit reusable objects at any time independent of the current open project. For example, if you open a new project, you can go to the object library, open a data flow, and edit it. The object will remain "dirty" (that is, your edited changes will not be saved) until you explicitly save it.

Functions are reusable objects that are not available in the object library. The software provides access to these objects through the function wizard wherever they can be used.

Some objects in the object library are not reusable in all instances:

- Datastores are in the object library because they are a method for categorizing and accessing external metadata.
- Built-in transforms are "reusable" in that every time you drop a transform, a new instance of the transform is created.

"Saving" a reusable object means storing the language that describes the object to the repository. The description of a reusable object includes these components:

- Properties of the object
- Options for the object
- Calls this object makes to other objects
- Definition of single-use objects called by this object

If an object contains a call to another reusable object, only the call to the second object is saved, not changes to that object's definition.

The description is stored even if the object is not successfully validated.

Objects are saved without prompting you:

- When you import an object into the repository.
- When you finish editing:
 - Datastores
 - Flat file formats
 - Nested schemas, such as DTD format, JSON format, or XML Schema

You can explicitly save the reusable object currently open in the workspace by choosing [Save](#) from the [Project](#) menu. If a single-use object is open in the workspace, the [Save](#) command is not available.

To save all objects in the repository that have changes, choose [Save All](#) from the [Project](#) menu.

You are prompted to save all objects that have changes when you execute a job and when you exit the Designer.

5.1.2 Single-use objects

Single-use objects appear only as components of other objects.

They operate only in the context in which they were created.

"Saving" a single-use object means storing the language that describes the object to the repository. The description of a single-use object can only be saved as part of the reusable object that calls the single-use object.

The description is stored even if the object does not validate.

5.2 Object options, properties, and attributes

Each object is associated with a set of options, properties, and attributes.

The following object information is available:

Object information	Description
Options	Controls the operation of an object. For example, in a datastore, an option is the name of the database to which the datastore connects.
Properties	Documents an object. For example, properties include the name and description of an object, the creation date, and so on. Properties describe an object, they do not affect an object's operation.
Attributes	Provides additional information about an object. Attribute values may also affect an object's behavior. To view attributes, open the Properties window and select the Attributes tab.

5.3 Descriptions of objects

Understand the purpose of each Data Services object including the object class by reading about the object in the following table.

Object descriptions

Object	Object class	Description
Annotation	Single-use	Note that describes a flow, part of a flow, or a diagram in the workspace that you attach to a workspace diagram.
Batch Job	Reusable	Defines activities that the software executes at a given time, and includes error, monitor and trace messages. You can add jobs only to a project. The batch job that you create is a direct reference to the batch job object in the object library. Only one reference to a job can exist in a project at one time.
Catch	Single-use	Specifies the steps to execute if an error occurs in a given exception group while a job is running.
COBOL copybook file format	Reusable	Defines the format for a COBOL copybook file source.
Conditional	Single-use	Specifies the steps to execute based on the result of a condition.

Object	Object class	Description
Data flow	Reusable	Specifies the requirements for extracting, transforming, and loading data from sources to targets. A data flow can be a part of a batch job or a real-time job.
Datastore	Reusable	Specifies the connection information needed to access a database or other data source. Cannot be dropped.
Document	Reusable	Available in certain adapter datastores, documents are data structures that can support complicated nested schemas.
DTD	Reusable	A description of an XML file or message. Indicates the format an XML document reads or writes.
Excel workbook format	Reusable	Defines the format for an Excel workbook source.
File format	Reusable	Indicates how flat file data is arranged in a source or target file.
File location	Reusable	Defines the file transfer protocol to use for transferring data files, including information about the remote and local servers. You associate a file location object to a format source or target in a data flow. As a source, the software uses the file location object information to transfer source data from a remote server to a local server. As a target, the software uses the file location object information to transfer the output file from the local server to the remote server.
Function	Reusable	Returns a value.
HDFS file format	Reusable	Describes the structure of a Hadoop distributed file system.
JSON file	Single-use	A batch or real-time source or target. As a source, a JSON file translates incoming JSON-formatted data into data that the software can process. As a target, a JSON file translates the data produced by a data flow, including nested data, into a JSON-formatted file.
JSON message	Single-use	A real-time source or target. As sources, JSON messages translate incoming JSON-formatted requests into data that a real-time job can process. As targets, JSON messages translate the result of the real-time job, including hierarchical data, into an JSON-formatted response and sends the messages to the Access Server.
Log	Single-use	Records information about a particular execution of a single job.
Message function	Reusable	Available in certain adapter datastores, message functions can accommodate XML messages when properly configured.
Nested Schemas template	Single-use	A target that creates a JSON file or an XML file that matches a particular input schema. No DTD, JSON Schema, or XML Schema is required.


Object	Object class	Description
Outbound message	Reusable	Available in certain adapter datastores, outbound messages are XML-based, hierarchical communications that real-time jobs can publish to adapters.
Project	Single-use	Groups jobs for convenient access.
Query transform	Single-use	Retrieves a data set that satisfies conditions that you specify.
Real-time job	Reusable	<p>Defines activities that the software executes on-demand.</p> <p>Real-time jobs are created in the Designer, then configured and run as services associated with an Access Server in the Administrator. Real-time jobs are designed according to data flow model rules and run as a request-response system.</p>
Script	Single-use	Evaluates expressions, calls functions, and assigns values to variables.
Source	Single-use	An object from which the software reads data in a data flow.
Table	Reusable	<p>Indicates an external DBMS table for which metadata has been imported, or the target table into which data is or has been placed.</p> <p>A table is associated with its datastore; it does not exist independently of a datastore connection. A table retrieves or stores data based on the schema of the table definition from which it was created.</p>
Target	Single-use	An object in which the software loads extracted and transformed data in a data flow.
Template table	Reusable	<p>A new table you want added to a database.</p> <p>All datastores except SAP datastores have a default template that you can use to create any number of tables in the datastore.</p> <p>The software creates the schema for each instance of a template table at runtime. The created schema is based on the data loaded into the template table.</p>
Transform	Reusable	<p>Performs operations on data sets.</p> <p>Requires zero or more data sets; produces zero or one data set (which may be split).</p>
Try	Single-use	Introduces a try/catch block.
While loop	Single-use	Repeats a sequence of steps as long as a condition is true.
Work flow	Reusable	Orders data flows and operations supporting data flows.

Object	Object class	Description
XML file	Single-use	A batch or real-time source or target. As a source, an XML file translates incoming XML-formatted data into data that the software can process. As a target, an XML file translates the data produced by a data flow, including nested data, into an XML-formatted file.
XML message	Single-use	A real-time source or target. As sources, XML messages translate incoming XML-formatted requests into data that a real-time job can process. As targets, XML messages translate the result of the real-time job, including hierarchical data, into an XML-formatted response and sends the messages to the Access Server.
XML Schema	Reusable	A description of an XML file or message. Indicates the format an XML document reads or writes.

5.3.1 Annotation

Use an annotation as a note that describes aspects of the data flow, part of the data flow, or a diagram in your workspace.

Characteristics of an annotation

Characteristic	Description
	Annotation icon
Class	Single-use
Access	Click the annotation icon in the tool palette, then click in the workspace.
Description	Annotations describe a flow, part of a flow, or a diagram in a workspace. An annotation is associated with the job, work flow, or data flow where it appears. When you import or export that job, work flow, or data flow, you import or export associated annotations.


i Note

An annotation has no options or properties.

5.3.2 Batch Job

Batch jobs process an entire dataset or multiple datasets.

Characteristics of a batch job

Characteristic	Description
	Batch job icon.
Class	Reusable
Access	Access batch jobs in the following locations: <ul style="list-style-type: none">• In the object library, click the <i>Jobs</i> tab.• In the project area, right-click the applicable project and select <i>New Batch Job</i>.
Description	A batch job is a set of objects that you can schedule and execute together. Set batch jobs to run once, or on a schedule. To execute the steps of any object, the object must be part of a job.

[Batch job objects \[page 24\]](#)

Objects that you add to batch jobs determine the source, target, and the types of Data Services processes to perform.

[Batch job automatic recovery \[page 25\]](#)

Rerun a failed job from the point of failure instead of from the beginning of the job using automatic recovery

[Batch job attributes and properties \[page 26\]](#)

The software assigns built-in attributes to an object. You can view and sometimes change an object property through the job Properties.

[Execution Options \[page 27\]](#)

Use execution options to help capture and diagnose errors using log, auditing, statistics collection, or recovery options.

[Single instance job \[page 30\]](#)

Ensure that only one instance of a job runs at a time by enabling *Single instance job*.

[Trace properties \[page 31\]](#)

Trace properties are types of information that the software monitors during job execution, and writes to trace files.

5.3.2.1 Batch job objects

Objects that you add to batch jobs determine the source, target, and the types of Data Services processes to perform.

A batch job can contain the following objects:

- Data flows, which include the following objects: Sources, transforms, targets
- Work flows

- Scripts
- Conditionals
- Try/catch blocks
- While Loops

Parent topic: [Batch Job \[page 24\]](#)

Related Information

[Batch job automatic recovery \[page 25\]](#)

[Batch job attributes and properties \[page 26\]](#)

[Execution Options \[page 27\]](#)

[Single instance job \[page 30\]](#)

[Trace properties \[page 31\]](#)

5.3.2.2 Batch job automatic recovery

Rerun a failed job from the point of failure instead of from the beginning of the job using automatic recovery

Run batch jobs with automatic recovery enabled to automatically recover from jobs that do not execute successfully. During automatic recovery, Data Services retrieves the results from steps that were successfully completed in the previous run and executes all other steps. Specifically, the software retrieves results from the following types of steps:

- Work flows
- Data flows
- Script statements
- Custom functions (stateless type only)
- SQL function
- EXEC function
- get_env function
- rand function
- sysdate function
- systime function

Automatic recovery is a setting that you make when you execute the job. For more information about recovery mechanisms, see the *Designer Guide*.

Parent topic: [Batch Job \[page 24\]](#)

Related Information

[Batch job objects \[page 24\]](#)

[Batch job attributes and properties \[page 26\]](#)

[Execution Options \[page 27\]](#)

[Single instance job \[page 30\]](#)

[Trace properties \[page 31\]](#)

5.3.2.3 Batch job attributes and properties

The software assigns built-in attributes to an object. You can view and sometimes change an object property through the job Properties.

Attributes

Batch jobs have the following built-in attributes:

Attribute	Description
Name	The name that the job creator provided for the job. This name appears on the object in the object library and in the calls to the object.
Description	The description that the job creator provided for the job.
Date created	The creation date of the object that the software assigns using the system date.

Properties

Batch job properties determine the information collected and logged when running the job. Set the default properties that apply each time you run the job or set execution or runtime properties that apply for a particular run. Execution properties override default properties.

Perform the following steps to view object properties:

1. Select the job in the project area or in the object library in Designer.
2. Right-click on the job name and choose *Properties*.

The *Properties* dialog box opens. The *Properties* dialog box contains some view-only information and editable information in several tabs. The available tabs are based on the object type.

Batch job property tabs

Tab	Content
General	The name and description of the job.
Attributes	Lists the attributes and values of the job.
Class Attributes	Lists attribute types for the selected object.
Execution Options	Lists default execution options. Override these settings by making settings in the Execute dialog box when you execute the job.
Trace	Set default trace properties, or view the default trace properties.
Global Variables	View global variable values connected to the batch job. For complete information about global variables, see the <i>Designer Guide</i> .

Parent topic: [Batch Job \[page 24\]](#)

Related Information

[Batch job objects \[page 24\]](#)

[Batch job automatic recovery \[page 25\]](#)

[Execution Options \[page 27\]](#)

[Single instance job \[page 30\]](#)

[Trace properties \[page 31\]](#)

5.3.2.4 Execution Options

Use execution options to help capture and diagnose errors using log, auditing, statistics collection, or recovery options.

Log information goes to one of four files in the `<DS_COMMON_DIR>\log\<Job_Server_Name>\<repository name>` directory:

- Error log file
- Monitor log file
- Trace log file
- Debug package file

When you execute a job, you can also select a system configuration and a Job Server or server group from the [Execution Options](#) tab of the [Execution Properties](#) window.

Select the *Execution Options* tab to set the following options.

Options	Description
<i>Collect statistics for monitoring</i>	<p>Select to display cache statistics in the Performance Monitor, located in the Management Console Administrator. Default setting: Not selected.</p> <div> <p>i Note</p> <p>Use this option only to look at the cache size.</p> </div> <p>For more information, see the Using Caches section in the <i>Performance Optimization Guide</i>.</p>
<i>Collect statistics for optimization</i>	<p>Select to collect statistics that the Data Services optimizer uses to choose an optimal cache type: In-memory and pageable. Default setting: Not selected.</p> <p>For more information, see the <i>Performance Optimization Guide</i>.</p>
<i>Disable data validation statistics collection</i>	<p>Select when you do not want to collect data validation statistics for any validation transforms in the job. Default setting: Not selected.</p> <p>For more information about data validation statistics, see the <i>Management Console Guide</i>.</p>
<i>Distribution level</i>	<p>Select the level with which to distribute job processing to multiple job servers. Options include the following:</p> <ul style="list-style-type: none"> • <i>Job</i>: The whole job executes on an available Job Server. • <i>Data flow</i>: Each data flow within the job executes on an available Job Server. • <i>Secondary data flow</i>: Each secondary data flow, either a separate transform or a function, within a data flow executes on an available Job Server. <p>For more information, see “Using grid computing to distribute data flows execution” and “Distributing Data Flow Execution” in the <i>Performance Optimization Guide</i>.</p>
<i>Enable auditing</i>	<p>Select to collect audit statistics for this specific job execution. Default setting: Selected.</p> <p>For more information about auditing, see the <i>Designer Guide</i>.</p>
<i>Enable recovery</i>	<p>Applies to batch jobs only. Select to enable the automatic recovery feature. When enabled, the software saves the results from completed steps and allows you to resume failed jobs. Default setting: Not selected.</p> <p>You cannot enable the automatic recovery feature when executing a job in data scan mode. Also, this option is available only as a runtime property. It is not available as a default property.</p> <p>For more information, see the <i>Designer Guide</i>.</p>
<i>Export Data Quality reports</i>	<p>Select to generate and export all specified job reports to the location specified in the <i>Report Server Configuration</i> node in the Management Console Administrator. Default setting: The reports are exported to <code><DS_COMMON_DIR>\DataQuality\reports\<repository\job></code>.</p>

Options	Description
Job Server or Server group	<p>Select either a Job Server or a server group to execute the job. The list contains Job Servers and server groups linked to the job repository.</p> <p>Define the Job Server with a host name and port. Define a server group by the server group name.</p> <p>When you select a Job Server or server group, remember that many objects in the Designer have options set relative to the Job Server location. For example:</p> <ul style="list-style-type: none"> • Directory and file names for source and target files • Bulk load directories <p>For more information, see the <i>Management Console Guide</i>.</p>
Monitor sample rate (# of seconds)	<p>Enter the number of seconds elapsed before the software writes information to the monitor log file and updates the job events. The software writes information about the status of each source, target, or transform. Default setting: 5 seconds.</p> <p>For example, if you enter 30, the logs update every 30 seconds.</p> <p>When you set the value, evaluate performance improvements gained by making fewer calls to the operating system against your ability to find errors quickly. With a higher monitor sample rate, more data collects before calling the operating system to open the file: performance improves. However, with a higher monitor rate, more time passes before you are able to see any errors.</p> <p>Setting the rate to 0 disables the monitoring feature.</p> <div> <p>Note</p> <p>If you use a virus scanner on your files, exclude the log from the virus scan. Otherwise, the virus scan analyzes the log repeated during the job execution, which causes a performance degradation.</p> </div>
Print all trace messages	<p>Select to print all trace messages to the trace log file for the current Job Server.</p> <p>Selecting this option overrides the trace properties set on the Trace tab.</p>
Recover from last failed execution	<p>Applies to batch jobs only. Select to resume a failed job. The software retrieves the results from any steps that were previously executed successfully and re-executes any other steps.</p> <p>This option is a runtime property and is only available if the job hasn't been executed or if you disabled recovery mode during a previous run.</p> <p>For more information about automatic recovery, see the <i>Designer Guide</i>.</p>
System configuration	<p>Select the system configuration to use when executing this job. A system configuration defines a set of datastore configurations, which define the datastore connections.</p> <p>If a system configuration is not specified, the software uses the default datastore configuration for each datastore.</p> <p>This option is a runtime property. This option is available only if there are system configurations defined in the repository.</p> <p>For more information about creating and managing multiple datastore configurations, see the <i>Designer Guide</i>.</p>

Options	Description
Use collected statistics	Select to have the optimizer use the cache statistics collected on a previous execution of the job. Default setting: Selected. For more information, see the <i>Performance Optimization Guide</i> .
Single instance	Select to prevent multiple instances of a job running.

Parent topic: [Batch Job \[page 24\]](#)

Related Information

[Batch job objects \[page 24\]](#)

[Batch job automatic recovery \[page 25\]](#)

[Batch job attributes and properties \[page 26\]](#)

[Single instance job \[page 30\]](#)

[Trace properties \[page 31\]](#)

5.3.2.5 Single instance job

Ensure that only one instance of a job runs at a time by enabling [Single instance job](#).

You can prevent multiple instances of a job running by selecting the [Single instance](#) checkbox in the [Properties](#) window. When [Single instance](#) is enabled, the system aborts any additional instances of the job while the current instance is running. In the Management Console, on the [Batch Job Status](#) tab, a red X appears next to the second instance of the job.

Look at the job error log file to find out why the software aborted the job. If the software aborted the job because it was the second instance of the same job, there is an error message that includes the currently running job start time as well as the job trace log name. You can verify that the first instance of the job is running by searching the [Batch Job Status](#) window for the job. It's possible that the first instance of the job crashed and the system was not able to update the job status to "fail". In this case, abort the job.

To ensure that a job is a single instance:

1. Right-click the job in the project area or the object library and choose [Properties](#).
2. Open the [Execution Options](#) tab of the [Properties](#) window and select the [Single instance](#) checkbox.
3. Click [OK](#).

Task overview: [Batch Job \[page 24\]](#)

Related Information

[Batch job objects \[page 24\]](#)

[Batch job automatic recovery \[page 25\]](#)

[Batch job attributes and properties \[page 26\]](#)

[Execution Options \[page 27\]](#)

[Trace properties \[page 31\]](#)

5.3.2.6 Trace properties

Trace properties are types of information that the software monitors during job execution, and writes to trace files.

Use the [Trace](#) tab of the [Properties](#) window to select the information that SAP Data Services monitors and writes to the trace log file during a job. The software writes trace messages to the trace log associated with the current Job Server and writes error messages to the error log associated with the current Job Server.

To enable or disable a trace property, open the [Trace](#) tab and select [Yes](#) or [No](#) in the [Value](#) list for the applicable property.

The following table contains descriptions for each trace property in alphabetical order.

Trace	Description
Access Server Communication	<p>Writes messages exchanged between the Access Server and a service provider, including the following message types:</p> <ul style="list-style-type: none">• A registration message, which tells the Access Server that the service provider is ready.• A request to execute that the Access Server sends to the service provider.• A response from the service provider to the Access Server.• All requests from the Access Server to shut down.
Assemblers	<p>Writes messages for Substitution Parameter and SDK Transforms:</p> <ul style="list-style-type: none">• Substitution parameters: Writes trace messages such as the substitution configuration used and the value substituted for each substitution parameter.• SDK transforms: Writes transform-specific information specified in XML format. The transform-specific information can be hierarchical. The Assemblers trace message writes the XML extracted or assembled at runtime.
Audit Data	<p>Writes a message when auditing:</p> <ul style="list-style-type: none">• Collects a statistic at an audit point• Determines if an audit rule passes or fails

Trace	Description
Data Flow	<p>Writes a message when the data flow starts, when the data flow successfully finishes, or when the data flow terminates due to error.</p> <p>This trace also reports when the bulk loader starts, any bulk loader warnings occur, and when the bulk loader successfully completes.</p>
Debug Package	Creates a package containing the trace, monitor, and log files and the ATL for the job.
IDoc file reader	<p>Writes a message when reading IDoc files including:</p> <ul style="list-style-type: none"> • start reading the IDoc file • stop reading the IDoc file • result of the IDoc file validation
Memory Source	Writes a message for every row retrieved from the memory table.
Memory Target	Writes a message for every row inserted into the memory table.
Optimized data flow	For consulting and customer assurance use.
RFC Function	<p>Writes a message related to RFC calls including:</p> <ul style="list-style-type: none"> • start of RFC call • end of RFC call • a message for each record received from the software for the RFC call
Row	Writes a message when a transform imports or exports a row.
SAP Table Reader	<p>Writes messages from readers of SAP system tables including:</p> <ul style="list-style-type: none"> • start reading from table • stop reading from table • start of connection to SAP system where table is present • end of connection to SAP system
Scripts and Script Functions	<p>Writes a message when the software runs a script or calls a script function. Specifically, this trace links a message when:</p> <ul style="list-style-type: none"> • The script is called. Scripts can be started any level from the job level down to the data flow level. Additional (and separate) notation is made when a script is called from within another script. • A function is called by the script. • The script successfully completes.
Session	Writes a message when the job description is read from the repository, when the job is optimized, and when the job runs.

Trace	Description
SQL Functions	<p>Writes data retrieved before SQL functions:</p> <ul style="list-style-type: none"> • Every row retrieved by the named query before the SQL is submitted in the <code>key_generation</code> function • Every row retrieved by the named query before the SQL is submitted in the <code>lookup</code> function (but only if <code>PRE_LOAD_CACHE</code> is not specified). • When mail is sent using the <code>mail_to</code> function.
SQL Loaders	<p>Writes a message when the bulk loader:</p> <ul style="list-style-type: none"> • Starts • Submits a warning message • Completes successfully • Completes unsuccessfully, if you select the <i>Clean up bulk loader directory after load</i> option. Additionally, for Microsoft SQL Server and SAP ASE, writes when the SQL Server bulk loader: • Completes a successful row submission • Encounters an error <p>This instance reports all SQL that the software submits to the target database, including:</p> <ul style="list-style-type: none"> • When a <code>truncate table</code> command executes if the <i>Delete data from table before loading</i> option is selected. • Any parameters included in PRE-LOAD SQL commands • Before a batch of SQL statements is submitted • When a template table is created (and also dropped, if you turn on the <i>Drop/Create</i> option) • When a <code>delete from table</code> command executes if you turn on auto correct (Informix environment only). <p>This trace also writes all rows that the software loads into the target.</p>
SQL Only	<p>Use with <i>Trace the SQL Transforms</i> option, the <i>Trace SQL Readers</i> option, or the <i>Trace SQL Loaders</i> option to stop the writing of trace messages for data sent and received from the database.</p>
SQL Readers	<p>Writes the SQL query block that a script, Query transform, or SQL function submits to the system. Also writes the SQL results.</p>
SQL Transforms	<p>Writes a message (using the <code>Table_Comparison</code> transform) about whether a row exists in the target table that corresponds to an input row from the source table.</p> <p>The trace message occurs before submitting the query against the target and for every row retrieved when the named query is submitted (but only if caching is not turned on).</p>

Trace	Description
Stored Procedure	Writes a message when the software calls a stored procedure. The report includes: <ul style="list-style-type: none"> • When the stored procedure starts • The SQL query submitted for the stored procedure call • The value (or values) of the input parameter (or parameters) • The value (or values) of the output parameter (or parameters) • The return value (if the stored procedure is a stored function) • When the stored procedure finishes
Tables	Writes a message when a table is created or dropped. The message indicates the datastore to which the created table belongs and the SQL statement used to create the table.
Trace ABAP	Writes a message when an ABAP data flow starts or stops, and to report the ABAP job status.
Trace Parallel Execution	Writes messages describing how data in a data flow is parallel processed.
Transform	Writes a message when a transform starts, completes, or terminates.
Work Flow	Writes a message when the work flow description is read from the repository, when the work flow is optimized, when the work flow runs, and when the work flow ends.

Parent topic: [Batch Job \[page 24\]](#)

Related Information

[Batch job objects \[page 24\]](#)

[Batch job automatic recovery \[page 25\]](#)


[Batch job attributes and properties \[page 26\]](#)

[Execution Options \[page 27\]](#)

[Single instance job \[page 30\]](#)

5.3.3 Catch

Catch is the second part of a try-catch block, which provides an alternative workflow if one or more errors occur in a job.

Characteristic	Description
	Catch icon
Class	Single-use
Access	With a work flow in the workspace, click the catch icon in the tool palette and then click in the workspace.
Description	A catch object is part of a serial sequence called a try/catch block. The try/catch block allows you to specify alternative work flows if one or more errors occur while executing a job. Try/catch blocks “catch” exception groups of errors, apply solutions that you provide, and continue execution.

[Catch best practices \[page 35\]](#)

Use these best practice suggestions to set up a successful try/catch block in your job setup.

[Catch attributes and exceptions \[page 36\]](#)

A catch object has one attribute but many exception groups that you can catch in a try/catch block.

[Catch error functions \[page 37\]](#)

Use catch functions inside the catch object to identify details about the error.

[Catch scripts \[page 38\]](#)

Catch scripts define actions for specific errors using catch functions.

5.3.3.1 Catch best practices

Use these best practice suggestions to set up a successful try/catch block in your job setup.

For each catch object in the try/catch block, specify the following:

- One or more groups of exceptions that the catch object handles.

i Note

If you want to assign different actions to different exception groups, add a catch for each set of actions.

- The actions to execute when an exception in the indicated exception groups occurs.
Optional but recommended: Define, test, and save the actions as a separate object rather than constructing them inside the catch editor. The actions can be a single script object, a data flow, a work flow, or a combination of these objects.
- Optional error functions inside the catch block to identify details of the error.

If an exception is thrown during the execution of a try/catch block, and if no catch object is looking for that exception group, then the block handles the exception with normal error logic.

For batch jobs only, do not reference output variables from a try/catch block in any subsequent steps if you are using the automatic recovery feature. Referencing such variables could alter the results during automatic recovery.

You can use try/catch blocks in any real-time job component. However, try/catch blocks cannot straddle a real-time processing loop and the initialization or cleanup component of a real-time job.

Parent topic: [Catch \[page 35\]](#)

Related Information

[Catch attributes and exceptions \[page 36\]](#)

[Catch error functions \[page 37\]](#)

[Catch scripts \[page 38\]](#)

5.3.3.2 Catch attributes and exceptions

A catch object has one attribute but many exception groups that you can catch in a try/catch block.

Catch objects have a name attribute. The object name appears on the object in the diagram.

Each try/catch exception error group corresponds with an error group number. For example, the try/catch block groups execution errors under 1001 and database access errors under group 1002. You select the exception groups to catch when you create the catch object. The following table describes each exception group.

Try/catch block exception groups

Exception group	Group number	Description
Catch All Exceptions	All	All errors
Execution errors	1001	Errors in the job server
Database access errors	1002	Errors from the database server while reading data, writing data, or bulk loading to tables
Database connection errors	1003	Errors connecting to database servers
Flat file processing errors	1004	Errors processing flat files
File access errors	1005	Errors accessing local and FTP files
Repository access errors	1006	Errors accessing the repository
SAP Execution errors	1007	Errors from the SAP system
System resource exception	1008	Errors accessing operating system resources
SAP BW execution errors	1009	Errors from the SAP BW system
XML processing errors	1010	Errors processing XML files and messages
COBOL copybook errors	1011	Errors processing COBOL copybook files

Exception group	Group number	Description
Excel book errors	1012	Errors processing Excel books
Data Quality transform errors	1013	Errors processing Data Quality transforms

For more information about try/catch blocks, see the Designer Guide.

Parent topic: [Catch \[page 35\]](#)

Related Information

[Catch best practices \[page 35\]](#)

[Catch error functions \[page 37\]](#)

[Catch scripts \[page 38\]](#)

5.3.3.3 Catch error functions

Use catch functions inside the catch object to identify details about the error.

The following table describes error functions that you can use in the script that your catch work flow executes.

Note

You can call these error functions only inside a catch script, a user function, or in an audit script for a data flow. If you call these error functions in any other place, a validation error occurs.

Catch object error functions and descriptions

Catch error function	Return data type and size	Description
<code>error_timestamp()</code>	timestamp	Returns the timestamp of the caught exception.
<code>error_context()</code>	varchar 512	Returns the context of the caught exception. For example: Session datapreview_job Dataflow debug_DataFlow Transform Debug
<code>error_message()</code>	varchar 512	Returns the error message of the caught exception.
<code>error_number()</code>	int	Returns the error number of the caught exception.

Parent topic: [Catch \[page 35\]](#)

Related Information

[Catch best practices \[page 35\]](#)

[Catch attributes and exceptions \[page 36\]](#)

[Catch scripts \[page 38\]](#)

[Descriptions of built-in functions \[page 1039\]](#)

5.3.3.4 Catch scripts

Catch scripts define actions for specific errors using catch functions.

A script is the most common action that a catch executes for a thrown exception. The catch script can contain the following:

- Catch error functions and other function calls
- Nested try/catch blocks
- “if” statements to perform different actions for different exceptions

Syntax

The following code shows the syntax for a try/catch block within a script:

```
try
begin
  <steps>
end
catch(<integer_constants>)
begin
  <steps>
end
```

Where

`<steps>`

Contains catch error functions, other function calls, if statements, or other statements that you want to perform for an error in the specified exception group number.

`<integer_constants >`

Contains one or more exception group numbers that you want to catch.

Use a comma to separate exception group numbers. For example,

```
catch (1002, 1003)
```

Specify ALL to catch all exceptions:

```
catch (ALL)
```

Parent topic: [Catch \[page 35\]](#)

Related Information

[Catch best practices \[page 35\]](#)

[Catch attributes and exceptions \[page 36\]](#)


[Catch error functions \[page 37\]](#)

[Descriptions of built-in functions \[page 1039\]](#)

5.3.4 COBOL copybook file format

Create a COBOL copybook file format to describe the structure of a COBOL copybook file.

Characteristics of a COBOL copybook file format object

Characteristic	Description
	Icon for a COBOL copybook file format.
Class	Reusable
Access	In the object library, click the Formats tab.
Description	A COBOL copybook file format describes the structure defined in a COBOL copybook file (denoted with a .cpy extension). Store templates for file formats in the object library. Use the templates to define the file format of a particular source in a data flow. <div>i Note You cannot use a COBOL copybook file format template as a target in a data flow.</div>

For details about working with COBOL copybook file formats, see the *Designer Guide*.

[Import or Edit COBOL copybook format \[page 40\]](#)

Use separate editors in Data Services Designer when you create and edit a COBOL copybook format file.

[COBOL copybook source options \[page 46\]](#)

When you use the COBOL copybook as a source in a data flow, enter source information so that Data Services knows how to identify and process the COBOL copybook.

5.3.4.1 Import or Edit COBOL copybook format

Use separate editors in Data Services Designer when you create and edit a COBOL copybook format file.

Create a new COBOL Copybook

Create a new COBOL copybook format file by right-clicking the COBOL Copybooks node in the *Formats* tab of the object library and selecting *New*. The *Import COBOL Copybook* dialog box contains the following tabs:

- Format
- Data File
- Data Access

Edit a COBOL Copybook

Edit a COBOL copybook format by right-clicking the format object name in the *Format* tab of the object library and selecting *Edit*. The *Edit COBOL Copybook* dialog box contains the following tabs:

- Field ID
- Record Length Field

[Format tab \[page 41\]](#)

Define format options for the COBOL copybook file format when you create a new file format.

[Data File tab \[page 42\]](#)

Define the data file for the COBOL copybook file format.

[Data Access tab \[page 44\]](#)

Specify the data transfer protocol so Data Services can access the COBOL copybook data file.

[Field ID tab \[page 45\]](#)

Create rules for associating records to schemas when the imported COBOL copybook has multiple schemas.

[Record Length Field tab \[page 46\]](#)

Identify the field that contains the length of the schema record and set an offset value.

Parent topic: [COBOL copybook file format \[page 39\]](#)

Related Information

[COBOL copybook source options \[page 46\]](#)

5.3.4.1.1 Format tab

Define format options for the COBOL copybook file format when you create a new file format.

Set format options only when you create a new COBOL copybook file format. You cannot edit the options when you edit the COBOL copybook file format.

Option descriptions for the Format tab

Format option	Description
<i>File name</i>	Type the COBOL copybook file name, or browse to and select the file name, usually with a <code>.cpy</code> file extension. This file contains the schema definition. For added flexibility, enter a variable for this option.
<i>Expand OCCURS</i>	Select to import OCCURS groups with each field in one of the following views: <ul style="list-style-type: none">Expanded view: Get sequential suffixes, one for each repetition. For example, <code>fieldname_1</code>, <code>fieldname_2</code>, and so on.Collapsed view: Appears only once in the copybook schema. For a collapsed view, the output schema matches the OCCURS group definition. For each input record there are several output records. If a copybook contains more than one OCCURS group, select the Expand OCCURS option. The default is selected.
<i>Ignore redefines</i>	Select to ignore REDEFINES clauses.
<i>Source format</i>	Select the format of the copybook source code. Options include: <ul style="list-style-type: none"><i>Free</i>: All characters on the line can contain COBOL source code.<i>Smart mode</i>: The software attempts to determine whether the source code is in standard or free format. If this attempt does not produce the desired result, choose the appropriate source format, standard or free, manually for reimporting.<i>Standard</i>: The traditional IBM mainframe COBOL source format, where each line of code is divided into the following five areas:<ul style="list-style-type: none">sequence number (1-6)indicator area (7)area A (8-11)area B (12-72)comments (73-80)
<i>Source codes [start]</i>	Enter the start column number of the copybook source file to use during the import. Typical value for IBM mainframe copybook: <ul style="list-style-type: none">7 for standard source format0 for free source format
<i>Source codes [end]</i>	Enter the end column number of the copybook source file to use during the import. Typical value for IBM mainframe copybook: <ul style="list-style-type: none">72 for standard source format9999 for free source format
<i>Generate record number field</i>	Select to create a new field at the beginning of the schema that the software populates at run-time with the record number.

Parent topic: [Import or Edit COBOL copybook format \[page 40\]](#)

Related Information

[Data File tab \[page 42\]](#)

[Data Access tab \[page 44\]](#)

[Field ID tab \[page 45\]](#)

[Record Length Field tab \[page 46\]](#)

5.3.4.1.2 Data File tab

Define the data file for the COBOL copybook file format.

Option descriptions for the Data File tab

Data file option	Description
File Location	<p>Optional. Select the name of an existing file location object.</p> <p>A file location object contains file transfer protocol information (such as FTP or SFTP) and local and remote server information. The software uses this information to copy the COBOL copybook data file from a remote server to a local server. Then it accesses the file from the local folder to use the data as a source in a data flow.</p> <div>i Note When you choose a file location object for this option, the software disables the Directory option and the Data Access tab.</div>
Delete file after transfer	<p>Available when you select a file location object.</p> <ul style="list-style-type: none">• Check to delete the local copy of the data file after the software loads it as a source in a data flow.• Uncheck to save the local copy of the data file after the software loads it as a source in the data flow.
Directory	<p>Specify the directory that contains the COBOL copybook data file to import. For added flexibility, you can select a variable for this option. If you include a directory path here, then enter only the file name in the File name field. During design, you can specify a file in one of the following ways:</p> <ul style="list-style-type: none">• For a file located on the computer where the Designer runs, you can use the Browse button.• For a file located on the computer where the Job Server runs, Type the path to the file. Type an absolute path or a relative path, but the Job Server must be able to access it. <div>i Note If you enter a file location object name in File Location, this option is disabled.</div>

Data file option	Description
<i>File name</i>	<p>Type the name or browse to the COBOL copybook data file. You can use variables or wild cards (* or ?).</p> <p>If you leave <i>Directory</i> blank, then type the full path and file name here.</p>
<i>Type</i>	<p>Select the record format type.</p> <ul style="list-style-type: none"> <i>Fixed(F)</i> <i>Variable(V)</i>
<i>Has record length</i>	<p>Select an option when variable-length records in the data file contain information about the length of each record. Available only when you select <i>Variable(V)</i> for <i>Type</i>. The values are:</p> <ul style="list-style-type: none"> <i>2-byte</i> integer <i>2-byte followed by 0x0000</i> (integer followed by two 0 bytes) <i>4-byte</i> integer <i>None</i>—No length information at the beginning of each record
<i>Record size</i>	Enter the record length in bytes when you select <i>Fixed(F)</i> for <i>Type</i> . All records in the file have this length, padded if necessary.
<i>Record trailer length</i>	Enter the length of extra character padding in bytes at the end of each record.
<i>Has record mark</i>	Select when there is an extra byte in the beginning of each record data.
<i>Integer format</i>	<p>Select the method that the existing data file uses to store binary data:</p> <ul style="list-style-type: none"> <i>Big endian</i>: The most significant byte comes first <i>Little endian</i>: The least significant byte comes first
<i>Code page</i>	Select the character encoding of character data in the data file.
<i>Skip first</i>	Select the number of data records to skip before starting to process the file. The default is {none}. For added flexibility, you can enter a variable for this option.
<i>Read total</i>	Defines the number of records to read and process. The default is {no limit}. For added flexibility, you can enter a variable for this option.
<i>Low Value</i> <i>High Value</i>	<p>Select a hexadecimal value for a low value, high value, or both from the copybook. You can also use a variable to define a different value at runtime.</p> <ul style="list-style-type: none"> <i>Low Value</i>: Default is 0x40 <i>High Value</i>: Default is 0xFF
	<p>❖ Example</p> <p>If the source field is binary 0x40, enter a <i>Low Value</i> of 0x40 and select the action <i>Convert to NULL</i>. The result would be as follows for these data types:</p> <ul style="list-style-type: none"> Char—Character represented by 0x40 Packed decimal—NULL Binary—0x40
<i>Action</i>	<p>For Low or High Value, select one of the following actions:</p> <ul style="list-style-type: none"> <i>No conversion</i>: Reads the value as an ASCII character (default). <i>Convert to NULL</i>: Converts the given value to NULL. <i>Convert to 0</i>: Converts the given value to 0.

Parent topic: [Import or Edit COBOL copybook format \[page 40\]](#)

Related Information

[Format tab \[page 41\]](#)

[Data Access tab \[page 44\]](#)

[Field ID tab \[page 45\]](#)

[Record Length Field tab \[page 46\]](#)

[File location object \[page 171\]](#)

5.3.4.1.3 Data Access tab

Specify the data transfer protocol so Data Services can access the COBOL copybook data file.

Select to use either FTP or a custom file transfer protocol method.

→ Tip

Consider using a file location object instead of defining data access in the [Data Access](#) tab. A file location object allows greater variety of file transfer methods, and provides for editing outside of the data flow.

If you do not select the [FTP](#) or [Custom](#) checkboxes, the software assumes that the data file is on the same computer as the Job Server.

Data access option	Description
FTP	Select to use FTP to access the data file.
Host	Type the computer (host) name, fully qualified domain name, or IP address of the computer where the data file resides.
User	Type the FTP user name.
Password	Type the FTP user password.
Directory	Type or browse to the directory that contains the COBOL copybook data file to import. If you include a directory path here, then enter only the file name in the Name field.
File name	Type or browse to the COBOL copybook data file Name . You can use variables or wild cards (* or ?). If you leave Directory blank, then type a full path and file name here.
Custom	Select to use a custom executable to access the data file.
Executable	Type the name of the program to read data file.
User	Type the user name.
Password	Type the password.
Arguments	Include any custom program arguments.

Parent topic: [Import or Edit COBOL copybook format \[page 40\]](#)

Related Information

[Format tab \[page 41\]](#)

[Data File tab \[page 42\]](#)

[Field ID tab \[page 45\]](#)

[Record Length Field tab \[page 46\]](#)

[File location object \[page 171\]](#)

5.3.4.1.4 Field ID tab

Create rules for associating records to schemas when the imported COBOL copybook has multiple schemas.

The *Field ID* tab only appears in the *Edit COBOL copybook* dialog box when the imported COBOL copybook has multiple schemas.

Option descriptions for the Field ID tab

Field ID option	Description
<i>Use field <FIELD NAME> as ID</i>	Select to set a value for the field selected in the top pane. Clear to not set a value for that field.
<i>Edit</i>	Changes the selected value in the Values pane to editable text.
<i>Delete</i>	Deletes the selected value in the Values pane.
<i>Insert above</i>	Inserts a new value in the Values pane above the selected value.
<i>Insert below</i>	Inserts a new value in the Values pane below the selected value.

Parent topic: [Import or Edit COBOL copybook format \[page 40\]](#)

Related Information

[Format tab \[page 41\]](#)

[Data File tab \[page 42\]](#)

[Data Access tab \[page 44\]](#)

[Record Length Field tab \[page 46\]](#)

5.3.4.1.5 Record Length Field tab

Identify the field that contains the length of the schema record and set an offset value.

Option descriptions for the Record Length Field tab

Record Length Field column	Description
Schema	Identify the data schema in the copybook.
Record length field	Select a field (one per schema) that contains the record length from the dropdown list. The software populates the value based on the field you choose.
Offset	Enter a value that, when added to the value in the Record length field option, results in the total record length. The default value for the offset is 4.

Parent topic: [Import or Edit COBOL copybook format \[page 40\]](#)

Related Information

[Format tab \[page 41\]](#)

[Data File tab \[page 42\]](#)

[Data Access tab \[page 44\]](#)

[Field ID tab \[page 45\]](#)

5.3.4.2 COBOL copybook source options

When you use the COBOL copybook as a source in a data flow, enter source information so that Data Services knows how to identify and process the COBOL copybook.

Open the source editor in the data flow. The editor includes the following tabs:

- Source
- Field clauses
- Data File
- Data Access

[Source tab \[page 47\]](#)

Set options to control performance, error handling, and other processing information for the COBOL copybook source file.

[Field Clauses tab \[page 49\]](#)

Set attributes for a selected column in the COBOL copybook source file data layout, which consists of various clause types.

Parent topic: [COBOL copybook file format \[page 39\]](#)

Related Information

[Import or Edit COBOL copybook format \[page 40\]](#)

5.3.4.2.1 Source tab

Set options to control performance, error handling, and other processing information for the COBOL copybook source file.

Option descriptions for the Source tab

Source option	Description
Make port	Select to make the source table an embedded data flow port. For more information, see the Embedded Data Flows section in the <i>Designer Guide</i> .

Performance option descriptions

Source option	Description
Join rank	<p>Enter a non-negative integer to indicate the rank of the source relative to other tables and files joined in a data flow. The software joins sources with higher join ranks before joining sources with lower join ranks.</p> <p>Join rank specified in the Query transform editor FROM tab overrides any join rank specified in a source. For new jobs, specify the join rank only in the Query transform editor.</p> <p>Must be a non-negative integer. Default value is 0.</p> <p>For more information, see the Other Tuning Techniques section in the <i>Performance Optimization Guide</i>.</p>
Cache	<p>Indicate whether the software reads the required data from the source and loads it into memory or pageable cache. The software reads an inner source of a join for each row of an outer source. Therefore, best practice suggests that you cache a source when it is used as an inner source in a join.</p> <ul style="list-style-type: none">• Yes: The source is always cached unless it is the outer-most source in a join. This setting is the default.• No: The source is never cached. <p>Cache specified in the Query transform editor FROM tab overrides any cache specified in a source. For new jobs, specify the cache only in the Query transform editor.</p>

Error handling option descriptions

Source option	Description
Log data conversion warnings	Select Yes to include data type conversion warnings in the error log. The default is Yes. Select No to exclude data type conversion warnings in the error log.
Maximum warnings to log	Select a limit for the number of warnings the software logs. <div>i Note This option is available only if the Log data conversion warnings option set to Yes. Default is {no limit}.</div>

Include file name column option descriptions

Source option	Description
Include file name column	Keep the default setting of No to not identify the source file. This is the default setting. add a column that contains the source file name in the source output. Defaults to No . Select Yes and the software adds a column that contains the source file name in the source output. The software identifies the source file in the following situations: <ul style="list-style-type: none">You specified a wildcard character to read multiple source COBOL copybooks at one timeYou load from different source copybooks on different runs
Modify	Click to modify File name column and Column size .
File name column	If the file name is included, the name of the column that holds the source file name. Defaults to DI_FILENAME .
Column size	If the file name is included, the size (in characters) of the column that holds the source file name. Defaults to 100 . If the size of the file name column is not large enough to store the file name, truncation occurs from the left.
Include path	If the file name is included, determines whether to include the full path name of the source file. Defaults to No .

Parent topic: [COBOL copybook source options \[page 46\]](#)

Related Information

[Field Clauses tab \[page 49\]](#)

5.3.4.2.2 Field Clauses tab

Set attributes for a selected column in the COBOL copybook source file data layout, which consists of various clause types.

Option descriptions for the Field Clauses tab

Field clauses option	Description
Possible values	Enter a value or multiple values to force the software to process only rows that contain the specified value or values. Separate multiple values with the pipe character (). Click the ellipses button to open the smart editor.
Level	Level number assigned to the field in the source record definition. Values are 01-50.
Original name	Name of the field in the copybook.
Original picture	PICTURE clause of the field in the copybook.
Original usage	USAGE clause of the field in the copybook.
Min occurs	If this field is a part of an OCCURS group, the minimum number of occurrences for this field.
Max occurs	If this field is a part of an OCCURS group, the maximum number of occurrences for this field.
Occurs depending on	Specifies the repetition counter field name for the ODO (OCCURS DEPENDING ON).
Redefines	Specifies the name of another field that this one REDEFINES.
Sign separate	Specifies whether the sign is stored separately from the field value.
Sign position	Specifies whether the sign is LEADING or TRAILING.
Multiply by	Specifies whether the field needs to be scaled, multiplied or divided by a certain number. For example, if the field PICTURE clause is 9(5)P(3), the value of the field from the data file is multiplied by 1000.

Parent topic: [COBOL copybook source options \[page 46\]](#)

Related Information


[Source tab \[page 47\]](#)

[Smart editor \[page 299\]](#)

5.3.5 Conditional

Conditionals are single-use objects that you use to define if/then/else logic in your work flow.

Define a conditional by specifying a condition and two logical branches. Place conditionals in a work flow. For complete information about conditionals, see the *Designer Guide*.

Characteristic	Description
	Conditional icon.
Class	Single-use
Access	With a work flow diagram in the workspace, click the conditional icon in the tool palette and click in the workspace.
Description	A conditional implements if/then/else logic in a work flow.

[Conditional best practices \[page 50\]](#)

Follow these conditional best practices to create a successful conditional.

Related Information

[Reference Guide: Smart Editor \[page 299\]](#)

5.3.5.1 Conditional best practices

Follow these conditional best practices to create a successful conditional.

For each conditional, specify the following:

- **If:** A Boolean expression defining the condition to evaluate.
The expression evaluates to TRUE or FALSE. You can use constants, functions, variables, parameters, and standard operators to construct the expression.

i Note

Do not put a semicolon (;) at the end of your expression in the **If** text box.

- **Then:** A work flow to execute if the condition is TRUE.
- **Else:** A work flow to execute if the condition is FALSE.
This branch is optional.

The **Then** and **Else** branches of the conditional can be any steps valid in a work flow, including a call to an existing work flow.

Conditionals have the following attribute:

Attribute	Description
Name	The name of the object. This name appears on the object in the diagram.

For complete information about creating a conditional, see the *Designer Guide*.

Parent topic: [Conditional \[page 49\]](#)


Related Information

[Smart editor \[page 299\]](#)

5.3.6 Data flow

A data flow contains the objects that instruct the software to access, process, and load data.

Characteristics of a data flow

Characteristic	Description
	Data flow icon
Class	Reusable
Access	<ul style="list-style-type: none">• In the object library, click the Data Flows tab.• With a work flow diagram in the workspace, click the data flow icon in the tool palette.
Description	<p>A data flow extracts, transforms, and loads data.</p> <p>You can define parameters to pass values into the data flow. You can also define variables for use inside the data flow.</p> <p>When SAP Data Services executes data flows, it optimizes the extract, transform, and load requirements into commands to the DBMS and commands executed internally. Where it can, the software runs these operations in parallel.</p>

For complete information about data flows, see the *Designer Guide*.

[Data flow objects and attributes \[page 52\]](#)

A data flow can contain a variety of object types based on the operations that you perform on the data.

[Executing jobs only once \[page 53\]](#)

When a data flow appears in a job or work flow more than once, control whether the software executes only the first occurrence of the data flow.

[Parallel processing \[page 53\]](#)

For better performance, set the data flow to run transforms and functions in parallel threads.

[Caching data \[page 54\]](#)

Caching data improves performance when the data flow contains joins, groupings, sorting, lookup tables, comparisons, and so on.

[Operation codes \[page 55\]](#)

Operation codes describe the status of each row in each data set in data flows.

5.3.6.1 Data flow objects and attributes

A data flow can contain a variety of object types based on the operations that you perform on the data.

A data flow has three categories of objects. Each category is comprised of several object types.

Object categories	Object types
Sources	Files, tables, JSON files, JSON messages (real-time jobs only), XML files, XML messages (real-time jobs only), documents, or pre-defined template tables
Targets	Files, tables, JSON files, JSON messages (real-time jobs only), XML files, XML messages (real-time jobs only), outbound messages, documents, Nested Schemas templates, or template tables
Transforms	The Query transform is the most commonly used transform

→ Tip

The software generates SQL code for table sources in data flows. View the SQL code to find areas to improve your data flow design.

Data flows have several built-in attributes that you can view through the Properties dialog box. Right-click the name of the data flow and select *Properties* from the dropdown menu.

Data flow properties

Attribute	Description
Name	The name of the object. This name appears on the object in the object library and in the calls to the object.
Description	Your description of the data flow.

If you delete a data flow from the object library, the software replaces all calls to the data flow with an icon that indicates the calls are no longer valid in the workspace.



Parent topic: [Data flow \[page 51\]](#)

Related Information

[Executing jobs only once \[page 53\]](#)

[Parallel processing \[page 53\]](#)

[Caching data \[page 54\]](#)

[Operation codes \[page 55\]](#)

5.3.6.2 Executing jobs only once

When a data flow appears in a job or work flow more than once, control whether the software executes only the first occurrence of the data flow.

Use this feature when you develop complex jobs with multiple paths. For example, if your job contains a Try-Catch block or conditionals and you want to ensure that Data Services executes a data flow only once.

Access the option in the *Properties* dialog box for the data flow. To access the *Properties* dialog box:

1. In the project area of Designer, expand the project and job in which the data flow resides.
2. Right-click the name of the data flow.
3. Select *Properties* from the dropdown list.

The *Execute only once* checkbox appears in the *General* tab after the data flow name.

Data Services executes only the first occurrence of the data flow and skips subsequent occurrences of the data flow in the job. However, under certain circumstances, the *Execute only once* option does not function as expected:

- **Parallel processing:** When you set a data flow that is marked to execute only once to run in parallel, the software only executes the first occurrence of the data flow. You cannot control which data flow the software executes first.
Only one data flow set to execute only once can execute in a single job. Therefore, the software skips subsequent occurrences of the data flow and generates the following trace message for each subsequent occurrence: Data flow <name> did not run more than one time. It is an execute only once flow.
- **Options that override the setting:** There are two options that override the *Execute only once* option set for a data flow in the *Properties* dialog box: The *Recover as a unit* work flow option and the *Enable recovery* job option.

Parent topic: [Data flow \[page 51\]](#)

Related Information

[Data flow objects and attributes \[page 52\]](#)

[Parallel processing \[page 53\]](#)

[Caching data \[page 54\]](#)

[Operation codes \[page 55\]](#)

5.3.6.3 Parallel processing

For better performance, set the data flow to run transforms and functions in parallel threads.

Set the software to run transforms in parallel by setting the option *Degree of parallelism* in your data flow *Properties*. The number that you set replicates transforms in the data flow which run as separate threads when the Job Server processes the data flow.

The number that you set is the maximum number of instances that the software can generate for each transform or function in the data flow.

For complete information about using the parallel processing, see the *Performance Optimization Guide*.

Parent topic: [Data flow \[page 51\]](#)

Related Information

[Data flow objects and attributes \[page 52\]](#)

[Executing jobs only once \[page 53\]](#)

[Caching data \[page 54\]](#)

[Operation codes \[page 55\]](#)

5.3.6.4 Caching data

Caching data improves performance when the data flow contains joins, groupings, sorting, lookup tables, comparisons, and so on.

Set the data flow to cache data in the data flow *Properties* dialog box by setting the *Cache type* option.

Cache type	Description
In Memory	Use in-memory cache if your data flow processes a small amount of data that can fit in the available memory.
Pageable	<p>Use pageable cache when your data flow processes a large amount of data that does not fit in memory. When memory-intensive operations such as Group By and Order By, exceed available memory, the software uses pageable cache to complete the operation.</p> <p>This setting is the default.</p> <div>i Note You cannot use pageable cache with nested data or LONG data types.</div>

Parent topic: [Data flow \[page 51\]](#)

Related Information

[Data flow objects and attributes \[page 52\]](#)

[Executing jobs only once \[page 53\]](#)

[Parallel processing \[page 53\]](#)

[Operation codes \[page 55\]](#)

5.3.6.5 Operation codes

Operation codes describe the status of each row in each data set in data flows.

The operation codes indicate how each row in the data set would be applied to a target table. The operation codes are as follows:

Operation Code	Description
NORMAL	<p>Creates a new row in the target.</p> <p>All rows in a data set are flagged as NORMAL when they are extracted by a source table or file. If a row is flagged as NORMAL when loaded into a target table or file, it is inserted as a new row in the target.</p> <p>Most transforms operate only on rows flagged as NORMAL.</p>
INSERT	<p>Creates a new row in the target.</p> <p>The Table_Comparison transform can flag rows with INSERT. The Insert flag indicates that a change occurred when it compares the data set to an earlier version of the same data set. The Map_Operation transform can also flag rows with INSERT.</p> <p>Only the History_Preserving and Key_Generation transforms accept data sets with rows flagged as INSERT as input.</p>
DELETE	<p>Ignores the row and does not load it.</p> <p>The Map_Operation and Table Comparison transforms can flag rows as DELETE.</p> <p>Only the History_Preserving transform with the <i>Preserve delete row(s) as update row(s)</i> option selected accept data sets with rows flagged as DELETE.</p>
UPDATE	<p>Overwrites an existing row in the target table.</p> <p>The Table_Comparison transform can flag rows with UPDATE. The Update flag indicates that a change occurred when it compares the data set to an earlier version of the same data set. The Map_Operation transform can also produce rows flagged as UPDATE.</p> <p>Only History_Preserving and Key_Generation transforms accept data sets with rows flagged as UPDATE as input.</p>

Parent topic: [Data flow \[page 51\]](#)

Related Information

[Data flow objects and attributes \[page 52\]](#)

[Executing jobs only once \[page 53\]](#)


[Parallel processing \[page 53\]](#)

[Caching data \[page 54\]](#)

5.3.7 Datastore

Datastores contain connection information to a data source such as a database, application, Web service, or an adapter.

Characteristics of a datastore

Characteristic	Description
	Datastore icon.
Class	Reusable
Access	In the object library, click the Datastores tab.
Description	A datastore provides a connection to a data source such as a database. Through the datastore connection, SAP Data Services can import descriptions of the data source such as its metadata. When you specify tables as sources or targets in a data flow, the software uses the datastore to determine how to read data from or load data to those tables. In addition, some transforms and functions require a datastore name to qualify the tables they access.

There are several types of datastores. For example, create a datastore to connect to a supported database, create a datastore to connect to a specific application such as SAP applications, or create a datastore to connect to an application using an adapter. The *Designer Guide* contains descriptions for all datastore types.

[Datastore editor \[page 57\]](#)

The datastore editor contains all options to create and edit all types of datastores.

[Application datastores \[page 66\]](#)

Configure an application datastore with connection information to the application.

[Database datastores \[page 68\]](#)

Create a database datastore so that SAP Data Services can read from and write to supported database types.

[Adapter datastores \[page 127\]](#)

Adapter datastores provide access to the application data that is accessed through the adapter.

[Web service datastores \[page 128\]](#)

Web service datastores represent a connection from Data Services to an external web service-based data source.

5.3.7.1 Datastore editor

The datastore editor contains all options to create and edit all types of datastores.

Create a new datastore by right-clicking in the [Datastores](#) tab of the object library and selecting [New](#). Right-click the name of an existing datastore and select [Edit](#) to edit that datastore.

- When you create a new datastore, the [Create New Datastore](#) window opens.
- When you edit a datastore, the [Edit Datastore](#) window opens.

The configuration options that you complete in the datastore editor change based on the type of datastore you choose. For example, if you create a database datastore, the option to choose a database type opens. After you select the database type, more applicable options appear. The options in the Advanced group are also determined by the datastore type and database type that you choose.

[Datastore properties \[page 58\]](#)

Datastore properties contain information such as name and description.

[Create and edit datastores \[page 58\]](#)

Create and edit datastores in the datastore editor.

[Basic and advanced configuration options \[page 59\]](#)

When you create a datastore, you start with basic settings and continue configuring the datastore using the advanced settings.

[Additional datastore configurations \[page 60\]](#)

Additional datastore configurations enable you to consolidate separate datastore connections into one source or target datastore.

[Datastore configuration in dataflows \[page 62\]](#)

When you use a multi configuration datastore in a dataflow, use caution when you change configurations.

[Importing database links \[page 65\]](#)

Use this datastore option to import and configure a database link in the Designer.

[Configurations with different database types \[page 65\]](#)

When your configurations are for different database types, options that don't apply to a type are not enabled.

[Working with Aliases \[page 66\]](#)

An alias is a logical owner name that you can create for objects that you use in different database environments.

Parent topic: [Datastore \[page 56\]](#)

Related Information

[Application datastores \[page 66\]](#)

[Database datastores \[page 68\]](#)

[Adapter datastores \[page 127\]](#)

[Web service datastores \[page 128\]](#)

5.3.7.1.1 Datastore properties

Datastore properties contain information such as name and description.

Datastore properties descriptions

Property	Description
Name	The name of the object. This name appears on the object in the object library and in the calls to the object. You cannot change the name of a datastore after creation.
Description	Text that you enter to describe and document the datastore.
Date_created	The date that you created the datastore. You cannot change this value.

i Note

If you delete a datastore from the object library, remove references to the datastore from the following locations:

- Source or target tables that use the datastore in work flows
- The lookup and key_generation functions and Key_Generation, History_Preserving, Table_Comparison, and SQL transform references

Parent topic: [Datastore editor \[page 57\]](#)

Related Information

[Create and edit datastores \[page 58\]](#)

[Basic and advanced configuration options \[page 59\]](#)

[Additional datastore configurations \[page 60\]](#)

[Datastore configuration in dataflows \[page 62\]](#)

[Importing database links \[page 65\]](#)

[Configurations with different database types \[page 65\]](#)

[Working with Aliases \[page 66\]](#)

5.3.7.1.2 Create and edit datastores

Create and edit datastores in the datastore editor.

When you create a new datastore, enter a name for your datastore based on your naming conventions. Then choose the datastore type from the drop-down list. The datastore type determine the options that appear next.

For example, choose Adapter as the datastore type, and the next two options that appear are: [JobServer](#) and [Adapter Instance Name](#). Choose a datastore type of Database, and the next option you set is [Database Type](#).

When you edit an existing datastore in the [Edit Datastore](#) window, you cannot change the datastore name, type, and type options. However, you can edit other basic information such as user name and password.

When you create or edit a datastore, click [Advanced](#) to open advanced configuration options.

Parent topic: [Datastore editor \[page 57\]](#)

Related Information

[Datastore properties \[page 58\]](#)

[Basic and advanced configuration options \[page 59\]](#)

[Additional datastore configurations \[page 60\]](#)

[Datastore configuration in dataflows \[page 62\]](#)

[Importing database links \[page 65\]](#)

[Configurations with different database types \[page 65\]](#)

[Working with Aliases \[page 66\]](#)

5.3.7.1.3 Basic and advanced configuration options

When you create a datastore, you start with basic settings and continue configuring the datastore using the advanced settings.

Settings for basic and advanced

Group of options	Descriptions
Basic datastore options	<p>Basic options are in the main page of the datastore editor.</p> <p>Enter a name for the datastore, then select a datastore type. Complete the remaining basic options that appear after you choose the datastore type. Options include security information, CDS (change data capture) options (when applicable), and DSN (data source name) options (when applicable).</p>
Advanced options	<p>The advanced options are based on the datastore type or the database type (when applicable).</p> <p>Advanced options can include FTP details, connection information, general loading information, and security information.</p> <p>Access advanced options by clicking Advanced in the configuration editor.</p>

Parent topic: [Datastore editor \[page 57\]](#)

Related Information

[Datastore properties \[page 58\]](#)

[Create and edit datastores \[page 58\]](#)

[Additional datastore configurations \[page 60\]](#)

[Datastore configuration in dataflows \[page 62\]](#)

[Importing database links \[page 65\]](#)

[Configurations with different database types \[page 65\]](#)

[Working with Aliases \[page 66\]](#)

[Database datastores \[page 68\]](#)

[Application datastores \[page 66\]](#)

5.3.7.1.4 Additional datastore configurations



Additional datastore configurations enable you to consolidate separate datastore connections into one source or target datastore.












To create additional configurations, click [Edit](#) after the Advanced options group to open the [Configurations for Datastore](#) dialog box.

The configuration editor contains a subset of the options in the datastore editor window. Each configuration appears in a column that contains the subset of options. The configuration editor also contains an editing toolbar. Use the configuration editor to add, edit, and remove datastore configurations.

The configurations editor always contains at least one configuration that reflects the first values that you set for the datastore. By default the software names the configuration Configuration 1. This first configuration is the default. When a datastore contains only one configuration, you cannot remove it from the datastore. All subsequent configurations appear as additional columns in the [Configurations for Datastore](#) dialog box.

The configurations editor provides a tool bar, which includes commands to add, edit, and remove configurations. Starting at the left of the tool bar, the buttons represent the following options:

Button icon	Option	Description
	Create New Configuration	Adds a new configuration with no values.
	Duplicate Configuration	<p>Creates a new configuration with identical settings as the selected configuration. The new configuration name must be unique, so the software uses the following naming convention: OldConfigurationName_Copy_CopyNumber.</p> <p>For example, if you duplicate a configuration called Test-Configuration, the software would name the duplicate Test-Configuration_Copy_1. If you do not rename the original or duplicate configuration and duplicate the original configuration again, the copy number appends by 1. So, the software would name the second duplicate TestConfiguration_Copy_2, and so on.</p>

Button icon	Option	Description
	Rename Configuration	Shifts input focus to the name of the selected configuration so you can edit it.
	Delete Configuration	Removes the configuration from the datastore and its column from the grid.
	Sort Configurations (Ascending)	Arranges the configurations by their names in ascending order. The arrangement is sensitive to the computer system locale.
	Sort Configurations (Descending)	Arranges the configurations by their names in descending order. The arrangement is sensitive to the computer system locale.
	Move Default to First	Moves the default configuration to the first column in the list. Does not change the order of other columns.
	Create New Alias	Adds a new alias name for the datastore. To map individual configurations to an alias, enter the real owner name of the configuration in the grid.
	Delete Alias	Removes the selected alias name for the datastore.
	Expand All Categories	Opens all the nodes so that every configuration property is visible.
	Collapse All Categories	Closes all the nodes so that every configuration property is hidden.
	Show/Hide Details	A toggle to show additional datastore options on the dialog box: Database type, Number of Configurations, and CDC status.
	Navigation box	This list contains the names of all configurations. Selecting a name from this list will (if necessary) scroll the configuration into view and highlight the configuration name in the grid. These commands (except for the Navigation box) also appear on a shortcut menu when you right-click any active cell on the grid.

You can also manage configurations by directly manipulating the grid.

- Rearrange the order of configuration columns by clicking a configuration name and dragging it left or right.
- Double-click a configuration name to edit it.
- Access one of the following options when you right-click a configuration name or any active cell on the grid:
 - [Add Linked Datastore](#)
 - [Delete Linked Datastore](#)
 - [Create New Alias](#)
 - [Delete Alias](#)

Using multiple configurations with database datastores can minimize your efforts to port existing jobs from one database type and version to another. The datastore editor supports quick creation of multiple configurations by allowing you to duplicate and rename configurations. Duplicating a configuration copies its options to create another configuration.

Each datastore must have only one Default configuration. When you select [Yes](#) to indicate a default configuration, the grid automatically sets the [Default configuration](#) value for the others to [No](#).

i Note

You can change the [Default configuration](#) value from [No](#) to [Yes](#), but you cannot change the value from [Yes](#) to [No](#). If you attempt to do so, the Designer displays an error message instructing you to select [Yes](#) for another configuration instead.

Parent topic: [Datastore editor \[page 57\]](#)

Related Information

[Datastore properties \[page 58\]](#)

[Create and edit datastores \[page 58\]](#)

[Basic and advanced configuration options \[page 59\]](#)

[Datastore configuration in dataflows \[page 62\]](#)

[Importing database links \[page 65\]](#)

[Configurations with different database types \[page 65\]](#)

[Working with Aliases \[page 66\]](#)

5.3.7.1.5 Datastore configuration in dataflows

When you use a multi configuration datastore in a dataflow, use caution when you change configurations.

When the dataflow contains the following objects, changing the datastore configuration in the dataflow can modify the dataflow language:

- Table targets
- Table transfer type used in Data_Transfer transform as a target
- SQL transforms

The software adds the target options and SQL transform text to additional datastore configurations based on the definitions in the existing configuration. The following table describes the target options that you can use when you add a new configuration to an existing datastore.

New configuration	Target options
Same database type and same or newer version as an existing configuration	Software automatically uses the values from an existing SQL transform, Target Table editor, or Data_Transfer transform editor values as applicable, including bulk loader options.
Different database type or the same database type but an older version as an existing configuration	You can select values from an existing database type and version by selecting the option from the Use values from list.

The *Use values from* list always contains the following options:

- Default values
- Database type and version for each configuration currently associated with the datastore

❖ Example

Your datastore contains two configurations: Oracle 9i and Microsoft SQL Server 2000. The datastore is connected to a data flow. Then you add a third configuration for DB2 to the datastore. The following default values appear in the *Use values from* list:

- Oracle 9i
- Microsoft SQL Server 2000

The default values are the same defaults that appear for all database targets, Data_Transfer target tables, and SQL transforms. Default SQL text is always blank. The following list contains some of the default target option values:

- *Row commit size* = 1000
- *Column comparison* = Compare by name
- *Delete data from table before loading* = not selected
- *Drop and re-create table* = not selected for regular tables (Selected for template tables)

If you select the *Restore values if they already exist* checkbox, which is selected by default, the software creates the new configuration. The software determines whether the SQL transform, target table editor, or Data_Transfer transform editor values already exist for the new database. If the database values already exist, the software restores the bulk load option. However, if no values exist for the database, the software sets the bulk load option to the default value of *None*.

Also, if you deselect *Restore values if they already exist*, the software sets the bulk load option to the default value of *None*.

❖ Example

You are working in a multi user environment and have a local datastore with configurations for Oracle 9i and SQL Server 2000. You also have existing dataflows that use target table from this datastore. You then delete Oracle 9i because you checked out a different version of the datastore from the central repository. Later, you want to add an Oracle 9i configuration to this datastore.

Deleting a version causes the software to remove the configuration, but not the target table, Data_Transfer target table, or SQL transform values. If you select *Restore values if they already exist* when you create a new configuration, the software determines whether values already exist for the database. If the software cannot find these values, the software uses values specified in the *Use values from* box.

After you click *Apply* to save a new configuration, the software:

- Copies any existing SQL transform, target table editor, and Data_Transfer target table editor values.
- Displays a report of the modified objects in a popup dialog box.
- Displays a report of the modified objects in the Output dialog box.

Parent topic: [Datastore editor \[page 57\]](#)

Related Information

[Datastore properties \[page 58\]](#)

[Create and edit datastores \[page 58\]](#)

[Basic and advanced configuration options \[page 59\]](#)

[Additional datastore configurations \[page 60\]](#)

[Importing database links \[page 65\]](#)

[Configurations with different database types \[page 65\]](#)

[Working with Aliases \[page 66\]](#)

[Modified object report contents \[page 64\]](#)

5.3.7.1.5.1 Modified object report contents

The software displays a modified object report after you add a new configuration to an existing datastore and click [Apply](#).

Description of the modified object report

Report column	Description
Dataflow	Names of the dataflows where language was modified.
Modified object	Objects in the dataflow that were affected.
Object type	Object types affected by the change. For example, table target or SQL transform.
Usage	Usage of the objects, such as source or target.
Has Bulk Loader	Whether the objects have a bulk loader.
Bulk Loader Copied	Whether the bulk loader option is copied.
Values Existed	Whether there are previous values.
Values Restored	Whether the previous values are restored.

You can use this report as a guide to manually change the values for options of targets, Data_Transfer target tables, and SQL transforms, as needed. In the popup dialog box that contains the modified objects report, you can sort results by clicking on column headers. You can also save the output to a file. The popup appears after you click [Apply](#) for each newly added configuration.

The software also clears and displays the results in the Output window after each newly added configuration. Because the datastore editor windows are modal, you cannot see the entire Output window or manipulate it. However, you can double-click one of the objects in the report to view the dataflow.

Related Information

[Datastore configuration in dataflows \[page 62\]](#)

5.3.7.1.6 Importing database links

Use this datastore option to import and configure a database link in the Designer.

A database link stores information about how to connect to a remote data source, such as its host name, database name, user name, password, and database type. The same information is stored in an SAP Data Services database datastore. You can associate the datastore to another datastore and then import an external database link as an option of a datastore. The datastores must connect to the databases defined in the database link.

For more information about database links and datastores, see the *Designer Guide*.

Parent topic: [Datastore editor \[page 57\]](#)

Related Information

[Datastore properties \[page 58\]](#)

[Create and edit datastores \[page 58\]](#)

[Basic and advanced configuration options \[page 59\]](#)

[Additional datastore configurations \[page 60\]](#)

[Datastore configuration in dataflows \[page 62\]](#)

[Configurations with different database types \[page 65\]](#)

[Working with Aliases \[page 66\]](#)

5.3.7.1.7 Configurations with different database types

When your configurations are for different database types, options that don't apply to a type are not enabled.

In the [Configurations for Datastore](#) dialog box, you can view multiple configurations at once, with each column representing a different configuration. When a datastore contains multiple configurations for different database types, some configurations have disabled cells based on the database type.

When an option does not apply to a configuration, the cell contains N/A and does not accept input. Some cells display in gray and do not accept input.

Parent topic: [Datastore editor \[page 57\]](#)

Related Information

[Datastore properties \[page 58\]](#)

[Create and edit datastores \[page 58\]](#)

[Basic and advanced configuration options \[page 59\]](#)

[Additional datastore configurations \[page 60\]](#)
[Datastore configuration in dataflows \[page 62\]](#)
[Importing database links \[page 65\]](#)
[Working with Aliases \[page 66\]](#)

5.3.7.1.8 Working with Aliases

An alias is a logical owner name that you can create for objects that you use in different database environments.

You can create an alias from the datastore editor for any datastore configuration. After you create an alias (for example, ALIAS1, ALIAS2), navigate horizontally to each configuration column. Enter the owner name to which that alias name maps or select an owner name from the dropdown list.

i Note

Data Services doesn't specifically label the owner name. The owner name displays in the grid, but it is in the row for the alias name. There is no row labeled Owner name.

When you delete an alias name, the delete operation also deletes each owner name applied for each configuration. The software removes the selected row, which includes the alias and all assigned owner names.

For instructions to create aliases, see the *Designer Guide*.

Parent topic: [Datastore editor \[page 57\]](#)

Related Information

[Datastore properties \[page 58\]](#)
[Create and edit datastores \[page 58\]](#)
[Basic and advanced configuration options \[page 59\]](#)
[Additional datastore configurations \[page 60\]](#)
[Datastore configuration in dataflows \[page 62\]](#)
[Importing database links \[page 65\]](#)
[Configurations with different database types \[page 65\]](#)

5.3.7.2 Application datastores

Configure an application datastore with connection information to the application.

The following table describes the applications available when you create an application datastore.

Descriptions of application datastores

Application	Datastore
Google BigQuery	<p>Connects to your BigQuery project to load data to existing Google tables and extract (read) data from existing tables to process in Data Services.</p> <ul style="list-style-type: none"> • Upload data generated by Data Services to existing tables in your Google BigQuery account for running queries. • Extract data from your Google BigQuery tables to use as a source in Data Services data flows. <p>Use with a Google BigQuery template as a target to automatically create the table in your Google account and upload processed data to the Google table.</p> <p>See the <i>Supplement for Google BigQuery</i> for more information.</p>
JDE OneWorld	<p>Connects to the database in your JDE OneWorld application. Datastore configuration options are based on which database type you select. The JDE OneWorld datastore type works with the following database types:</p> <ul style="list-style-type: none"> • DB2 • Microsoft SQL Server • ODBC • Oracle <p>For details about configuring the datastore options for JD Edwards applications, refer to the <i>Supplement for J.D. Edwards</i>.</p>
JDE World	<p>Connects to the ODBC database in your JDE World application.</p> <p>For details about configuring the datastore options for JD Edwards applications refer to the <i>Supplement for J.D. Edwards</i>.</p>
Oracle Applications	<p>Connects to data in your Oracle applications. For details about configuring the datastore options for Oracle applications, refer to the <i>Supplement for Oracle Applications</i>.</p>
PeopleSoft	<p>Connects to the following database types in your PeopleSoft application. Options vary based on the database type you select. Applicable PeopleSoft databases are:</p> <ul style="list-style-type: none"> • Microsoft SQL Server • Oracle <p>For more information about PeopleSoft applications, refer to the <i>Supplement for PeopleSoft</i>.</p>
Replication Server CDC	<p>Connects to changed-data capture (CDC) jobs using SAP PowerDesigner Data Movement model.</p>

Application	Datastore
SAP Applications	Connects to the applicable SAP application. Find descriptions for SAP application datastores in the <i>Supplement for SAP</i> .
SAP BW Source	
SAP BW Target	
SAP DQM Microservices	<p>Connects to SAP Data Quality Management, microservices for location data for use of the DQM Microservices transform within Data Services.</p> <p>Find datastore option descriptions for the DQM Microservices datastore in the <i>Supplement for SAP</i>.</p>
Siebel	<p>Connects to the following database types in your Siebel application:</p> <ul style="list-style-type: none"> • DB2 • Microsoft SQL Server • Oracle <p>For more information about Siebel applications, refer to the <i>Supplement for Siebel</i>.</p>
Web Service REST	Find complete option descriptions in the <i>Integrator Guide</i> .
Web Service SOAP	Find complete option descriptions in the <i>Integrator Guide</i> .

After you create a datastore, you can import metadata about the objects, such as tables and functions, into that datastore in the object library.

For complete information about adapter datastores, see the *Supplement for adapters*.

Parent topic: [Datastore \[page 56\]](#)

Related Information

[Datastore editor \[page 57\]](#)

[Database datastores \[page 68\]](#)

[Adapter datastores \[page 127\]](#)

[Web service datastores \[page 128\]](#)

5.3.7.3 Database datastores

Create a database datastore so that SAP Data Services can read from and write to supported database types.

Database datastores require specific connection information. The required information appears in the datastore editor when you choose the database type. See each supported database type description for specific connection information and other option requirements.

i Note

An option to use changed-data capture (CDC) is available for a subset of databases. Keep in mind that when you choose CDC, the advanced option groups in the datastore editor for *General*, *Bulk Loader*, and *FTP* do not display because a CDC datastore is read-only and you can only use it as a source.

- [Amazon Redshift \[page 70\]](#)
- [Attunity Connector \[page 85\]](#)
- [Data Federator \[page 86\]](#)
- [DB2 \[page 89\]](#)
- [HP Vertica datastore options \[page 91\]](#)
- [Impala connectivity using the Cloudera ODBC driver \[page 75\]](#)
- [Informix \[page 93\]](#)
- [Memory \[page 95\]](#)
- [Microsoft SQL Server \[page 95\]](#)
- [MySQL \[page 98\]](#)
- [Netezza \[page 101\]](#)
- [ODBC \[page 108\]](#)
- [Oracle \[page 109\]](#)
- [Persistent Cache \[page 112\]](#)
- [SAP ASE \[page 116\]](#)
- [SAP HANA datastore options \[page 112\]](#)
- [SAP SQL Anywhere \[page 114\]](#)
- [SAP Vora datastore \[page 121\]](#)
- [Sybase IQ \[page 118\]](#)
- [Teradata \[page 125\]](#)

Parent topic: [Datastore \[page 56\]](#)

Related Information

[Datastore editor \[page 57\]](#)

[Application datastores \[page 66\]](#)

[Adapter datastores \[page 127\]](#)

[Web service datastores \[page 128\]](#)

5.3.7.3.1 Enable Automatic Data Transfer

For automatic data transfer, the Data_Transfer transform uses transfer tables in the datastore to push down subsequent database operations for more efficient processing.

The Enable automatic data transfer check box is selected by default when you create a new datastore and you choose *Database* for *Datastore type*. The check box displays for all databases except Attunity Connector, Data Federator, Memory, and Persistent Cache.

Related Information

[Data_Transfer \[page 358\]](#)

5.3.7.3.2 Change data capture (CDC)

CDC updates database files with data that changed since the last refresh.

The software locates and extracts only data that changed since the last refresh, which can improve performance. Additionally, using CDC preserves history so you can track data changes over time.

The CDC option is active for specific data types and only for new database datastore creation.

Restrictions when you choose the CDC option:

- Any additional configurations for the datastore must be for the same database type.
- The datastore advanced option groups for *General*, *Bulk Loader*, and *FTP* are not available. That is because a CDC datastore is read-only so you can only use the datastore as a source in a data flow.

5.3.7.3.3 Amazon Redshift

A Redshift datastore enables you to read or load Redshift tables in a data flow, preview data, and more.

The following tables describe the Redshift options when you create or edit a datastore.

Main window options

Option	Description
Database Version	Select the Redshift database version from the dropdown list.

Option	Description
<i>Data Source Name</i>	<p>Select an existing data source name (DSN) from the drop-down list.</p> <div> <p>i Note</p> <p>You define the DSN with the Amazon Redshift ODBC driver in the ODBC Administrator.</p> </div>
<i>User Name</i>	Enter the user name of the account through which Data Services accesses the database.
<i>Password</i>	Enter the password associated with the <i>User Name</i> .
<i>Enable Automatic Data Transfer</i>	<p>Select to enable transfer tables in this datastore. The Data_Transfer transform uses transfer tables to push down subsequent database operations.</p> <p>This option is enabled by default.</p>
Advanced options	
Redshift option	Description
<i>Common</i>	
<i>Additional connection parameters</i>	<p>Enter information for any additional connection parameters. Use the following format: <code><parameter1=value1; parameter2=value2></code></p>
<i>General</i>	
<i>Rows per commit</i>	<p>Enter the maximum number of rows loaded to a target table before saving the data. The default is 1000.</p> <p>This value is the default commit size for target tables in this datastore.</p> <p>You can overwrite this value for individual target tables.</p>
<i>Bulk loader directory</i>	<p>Enter or browse to the location where the software writes data files for bulk loading. You can use a variable for this option.</p> <p>The default location for the bulk loader directory is <code><DS_COMMON_DIR>/log/bulkload</code>.</p>
<i>Overflow file directory</i>	Enter or browse to the location where the software writes overflow files written by target tables in this datastore. You can use a variable for this option.
<i>Sessions</i>	

Redshift option	Description
Additional session parameters	Additional session parameters specified as a valid SQL statement. Delimit multiple SQL statements with semicolons.
Aliases (Click here to create)	
Aliases	Click to open the Create New Alias popup dialog. Enter an alias name, click <i>OK</i> , and enter the owner name in the Alias text row.

Related Information

[Amazon S3 protocol options \[page 176\]](#)

[load_from_s3_to_redshift \[page 1136\]](#)

[Amazon Redshift data types \[page 319\]](#)

[Amazon Redshift source \[page 209\]](#)

[Amazon Redshift target table options \[page 226\]](#)

[Configuring Redshift as source using DSCONNECTIONMANAGER \[page 73\]](#)

[Enable Automatic Data Transfer \[page 70\]](#)

5.3.7.3.1 About Amazon Redshift datastores

Use an Amazon Redshift datastore to import and load tables, load Amazon S3 data files, and more.

Use a Redshift database datastore for the following tasks:

- Import tables
- Read or load Redshift tables in a data flow
- Preview data
- Create and import template tables
- Load Amazon S3 data files into a Redshift table using the built-in function `load_from_s3_to_redshift`

Use an Amazon Redshift ODBC driver to connect to the Redshift cluster database. The Redshift ODBC driver connects to Redshift on Windows and Linux platforms only.

For information about downloading and installing the Amazon Redshift ODBC driver, see the Amazon Redshift documentation on the Amazon website.

i Note

Enable secure socket layer (SSL) settings in the Amazon Redshift ODBC Driver. In the [Amazon Redshift ODBC Driver DSN Setup](#) window, set the [SSL Authentication](#) option to *allow*.

5.3.7.3.2 Configuring Redshift as source using DSCONNECTIONMANAGER

Use the DSCONNECTIONMANAGER to configure Amazon Redshift as a source for Data Services.

1. Download and install the Amazon Redshift ODBC driver for Linux. For more information, read about installing the Redshift ODBC driver for Linux in the *Amazon Redshift Management Guide* on the Amazon website (<http://docs.aws.amazon.com/redshift/latest/mgmt/install-odbc-driver-linux.html>).

After installing the ODBC driver on Linux, configure the following files:

- `amazon.redshiftodbc.ini`
- `odbc.ini`
- `odbcinst.ini`

For more information about configuring these `.ini` files, see the *Amazon Redshift Management Guide* on the Amazon website (<http://docs.aws.amazon.com/redshift/latest/mgmt/odbc-driver-configure-linux-mac.html>).

2. At the end of `/opt/amazon/redshiftodbc/lib/64/amazon.redshiftodbc.ini`, add a line to point to the `libodbcinst.so` file. This file is in the `unixODBC/lib` directory.

For example, `ODBCInstLib=/home/ec2-user/unixODBC/lib/libodbcinst.so`.

In addition, in the `[Driver]` section of the `amazon.redshiftodbc.ini` file, set `DriverManagerEncoding` to UTF-16.

For example,

```
[Driver]
DriverManagerEncoding=UTF-16
```

3. Configure the Linux ODBC environment.
 - a. Run `DSCONNECTIONMANAGER.sh` and configure a data source for Redshift.

Note

The Unix ODBC Lib Path is based on where you install the driver. For example, for Unix ODBC 2.3.4 the path would be `/build/unixODBC-232/lib`.

Specify the DSN name from the list or add a new one:

`DS42 REDSHIFT`

Specify the User Name:

`<name of the user>`

Type database password:(no echo)

Retype database password:(no echo)

Specify the Unix ODBC Lib Path:

`/build/unixODBC-232/lib`

Specify the Driver:

`/opt/amazon/redshiftodbc/lib/64/libamazonredshiftodbc64.so`

Specify the Driver Version:'8'

`8`

Specify the Host Name:

`<host name/IP address>`

Specify the Port:

`<port number>`

Specify the Database:

`<database name>`

```
Specify the Redshift SSL certificate verification mode
[require|allow|disable|prefer|verify-ca|verify-full]:'require'
require
Testing connection...
Successfully added database source.
```

Related Information

[Amazon Redshift data types \[page 319\]](#)
[Amazon Redshift source \[page 209\]](#)
[Amazon Redshift target table options \[page 226\]](#)
[Amazon S3 protocol options \[page 176\]](#)
[load_from_s3_to_redshift \[page 1136\]](#)
[Amazon Redshift \[page 70\]](#)

5.3.7.3.4 Apache Impala

Create a database datastore for Apache Impala, which is an open source database for Apache Hadoop.

Before you create an Apache Impala datastore, import the Cloudera driver and create a data source name (DSN). To create an Apache Impala datastore, open the datastore editor and select ODBC for the [Data Type](#). Then select the DSN you created for this datastore and complete the remaining options as applicable.

Use the datastore to import Impala tables, then use data from Impala as a source or target in a data flow.

Before you work with Apache Impala, be aware of the following limitations:

- Data Services supports Impala 2.5 and later.
- Data Services supports only Impala scalar data types. Data Services does not support complex types such as ARRAY, STRUCT, or MAP.

For descriptions of the Advanced options for creating the datastore, see “ODBC editor fields and options”.

[Impala connectivity using the Cloudera ODBC driver \[page 75\]](#)

Before you create an Impala database datastore, connect to Apache Impala using the Cloudera ODBC driver.

[Creating an Apache Impala datastore and DSN for Cloudera driver \[page 76\]](#)

For Windows, create the Apache Impala datastore and create a DSN for the Cloudera driver in Designer.

Related Information

[ODBC editor fields and options \[page 104\]](#)

5.3.7.3.4.1 Impala connectivity using the Cloudera ODBC driver

Before you create an Impala database datastore, connect to Apache Impala using the Cloudera ODBC driver.

Follow these high-level steps to connect to Apache Impala using the Cloudera ODBC driver. For more in-depth information, consult the Cloudera documentation.

1. Enable Impala Services on the Hadoop server.
2. Download and install the Cloudera ODBC driver (<https://www.cloudera.com/downloads/connectors/impala/odbc/2-5-26.html>):
 - For Windows, use `ClouderaImpalaODBC64.msi`.
 - For SUSE, use `ClouderaImpalaODBC-2.5.39.1020-1.x86_64.rpm`.
 - For RedHat 7, use `ClouderaImpalaODBC-2.5.39.1020-1.el7.x86_64.rpm`.
3. Configure an Impala data source name (DSN).

For Windows instructions, see [Creating an Apache Impala datastore and DSN for Cloudera driver \[page 76\]](#).

Linux:

- a. Run `DSConnectionManager.sh`. For more information about `DSConnectionManager`, see “Using the Connection Manager for UNIX systems” in the *Administrator Guide*.

Note

This example has Kerberos and Secure Sockets Layer (SSL) enabled.

```
The ODBC ini file is <path to the odbc.ini file>
There are available DSN names in the file:
[DSN name 1]
[DSN name 2]
Specify the DSN name from the list or add a new one:
imp_ssl_1
Specify the User Name:
<name of the user>
Type database password:(no echo)
Retype database password:(no echo)
Specify the Host Name:
<host name/IP address>
Specify the Port:'21050'
<port number>
Specify the Database:
default
Specify the Unix ODBC Lib Path:
The Unix ODBC Lib Path is based on where you install the driver.
For example, /build/unixODBC-2.3.2/lib.
Specify the Driver:
/opt/cloudera/impalaodbc/lib/64/libclouderaimpalaodbc64.so
Specify the Impala Auth Mech [0:noauth|1:kerberos|2:user|3:user-password]:'0'
1
Specify the Kerberos Host FQDN:
<hosts fully qualified domain name>
Specify the Kerberos Realm:
<realm name>
Specify the Impala SSL Mode [0:disabled | 1:enabled]:'0'
1
Specify the Impala SSL Server Certificate File:
```

```
<path to certificate.pem>
Testing connection...
Successfully added database source.
```

4. Create an ODBC datastore with the Impala DSN and then import the Impala tables.
5. Optional. Enable Kerberos authentication.

For Windows instructions, see [Creating an Apache Impala datastore and DSN for Cloudera driver \[page 76\]](#).

Linux:

- a. When configuring the Impala DSN in the Connection Manager (see step 3), enable Kerberos by setting *Specify the Impala Auth Mech* `[0:noauth|1:kerberos|2:user|3:user-password]`: to 1.
- b. Enter the *Kerberos Host FQDN*.
- c. Enter the *Kerberos Realm*.

i Note

DSConnectionManager does not test the Kerberos connection. It saves all the input to the \$ODBCINI file and tests the connection at runtime.

6. Optional. Enable Secure Socket Layer (SSL).

For Windows instructions, see [Creating an Apache Impala datastore and DSN for Cloudera driver \[page 76\]](#).

Linux:

- a. When configuring the Impala DSN in the Connection Manager (see step 3), enable SSL by setting *Specify the Impala SSL Mode* `[0:disabled|1:enabled]`: to 1.
- b. Enter the path to the Impala `certificate.pem` file.

Task overview: [Apache Impala \[page 74\]](#)

Related Information

[Creating an Apache Impala datastore and DSN for Cloudera driver \[page 76\]](#)

5.3.7.3.4.2 Creating an Apache Impala datastore and DSN for Cloudera driver

For Windows, create the Apache Impala datastore and create a DSN for the Cloudera driver in Designer.

Enable Impala Services in the Hadoop server. Download the Cloudera driver for your platform. For instructions, see [Impala connectivity using the Cloudera ODBC driver \[page 75\]](#).

Follow these steps to create an Apache Impala datastore and DSN with the Cloudera driver on Windows.

i Note

If you use Linux, follow the steps in [Impala connectivity using the Cloudera ODBC driver \[page 75\]](#) before you create the datastore. Linux users do not need to perform the steps to create the DSN for the Cloudera driver in the following steps.

1. Open the [Create New Datastore](#) editor in Designer.

For complete information about creating a new datastore, see the *Designer Guide*.

2. Select [Database](#) from the [Datastore Type](#) dropdown list.
3. Select [ODBC](#) from the [Database Type](#) dropdown list.

If you used the DSConnection Manager to create a DSN with the Cloudera driver, skip steps 4-9.

4. Click [ODBC Admin](#).

The [ODBC Data Source Administrator](#) opens.

5. Open the [System DSN](#) tab and select the Cloudera driver that you downloaded from the [System Data Sources](#) list. Click [Configure](#).

The [Cloudera ODBC Driver for Impala DSN Setup](#) dialog box opens.

6. Enter the required information based on your system, and click [Advanced Options](#).
7. In the [Advanced Options](#) window, enable [Use SQL Unicode Types](#). Close [Advanced Options](#).
8. Optional. Follow these steps to enable Kerberos authentication:
 - a. In the [Cloudera ODBC Driver for Impala DSN Setup](#) window, select [Kerberos](#) from the [Mechanism](#) dropdown list.
 - b. Enter the [Realm](#) name.

A realm is a set of managed nodes that share the same Kerberos database. For example, your realm name might be Cloudera.

- c. Enter the [Host FQDN](#), which is the fully qualified domain name of the Hive Server host.
 - d. Enter the [Service Name](#), which is the service principal name of the Hive server.
 - e. Enable the [Canonicalize Principal FQDN](#) option, which canonicalizes the host FQDN in the server service principal name.
9. Optional. Follow these steps to enable Secure Sockets Layer (SSL).
 - a. In the [Cloudera ODBC Driver for Impala DSN Setup](#) window, select [No Authentication \(SSL\)](#) from the [Mechanism](#) dropdown list.
 - b. Click [Advanced Options](#).
 - c. Enter or browse to the Cloudera certificate file in [Trusted Certificates](#).

The default path to the Impala `certificate.pem` file automatically populates.

After you have completed options for configuring the Cloudera driver and DSN, close the [Advanced Options](#), [Cloudera ODBC Driver for Impala DSN Setup](#), and the [ODBC Administrator](#).

10. Select the Cloudera DSN that you just created from the [Data Source Name](#) dropdown list.

The DSN appears in the dropdown list only if you created it with the DSConnection Manager, or with the ODBC Administrator.

11. Click [Advanced](#) in the [Create New Datastore](#) editor and complete the options as necessary.
12. Optional. To process multi byte data in Impala tables, go to the [Locale](#) group and set the [Code page](#) option to [utf-8](#).

13. In the *ODBC Date Function Support* group, set the *Week* option to *No*.

If you do not set this option to *No*, the result of the Data Services built-in function `week_in_year()` may be incorrect.

Task overview: [Apache Impala \[page 74\]](#)

Related Information

[Impala connectivity using the Cloudera ODBC driver \[page 75\]](#)

5.3.7.3.5 Hive datastores

Use a Hive datastore to access data from your Hive warehouse.

Hive is a data warehouse which facilitates reading, writing, and managing large datasets residing in distributed storage such as Hadoop Distributed File System (HDFS). To query Hive, use HiveQL, a type of SQL syntax.

Hive datastores use the Hive adapter to connect to your Hive warehouse. Data Services supports two types of Hive datastores:

- Hive adapter datastore: Data Services is installed within the Hadoop cluster. Configure the datastore without using a data source name (DSN) and a supported Hive ODBC driver.
- Hive database datastore: Data Services is installed on any machine. Configure the datastore using DSN and a supported Hive ODBC driver.

Use either Hive datastore type for the following tasks:

- Read from and write to the Hive server using Hive tables as source or target tables.
- Use a Hive template table in your data flow.
- Preview data from the Hive table.

Supported Hive ODBC Drivers

The Hive database datastore supports the following ODBC Drivers:

- Cloudera
- Hortonworks
- MapR

For more information about the specific driver versions currently supported, see the *Product Availability Matrix (PAM)*.

Limitations

- Operations such as DELETE, UPDATE, and UPINSERT are not natively supported by the Hive server.
- Parameterized SQL is not supported; the Hive server does not support the parameter marker.

[Hive adapter datastore configuration options \[page 79\]](#)

To use the Hive adapter, create a Hive adapter datastore in Data Services.

[Using an ODBC driver to connect to a remote Hive Server \[page 82\]](#)

Use a supported Hive ODBC driver to create a DSN (data source name) to remotely connect to your Hive server.

[Configuring bulk loading for Hive \[page 84\]](#)

Use a combination of Hadoop objects to configure bulk loading for Hive targets in a data flow.

5.3.7.3.5.1 Hive adapter datastore configuration options

To use the Hive adapter, create a Hive adapter datastore in Data Services.

The following table contains descriptions for the datastore configuration options that apply to the Hive adapter:

Option	Description
<i>Host name</i>	The name of the machine that is running the Hive service.
<i>Port number</i>	The port number of the machine that is running the Hive service.
<i>Username and Password</i>	<p>The user name and password associated with the adapter database to which you are connecting.</p> <p>If you are using Kerberos authentication, the user name should include the Kerberos realm. For example: <code>dsuser@BIGDATA.COM</code>. If you use Kerberos keytab for authentication, you do not need to complete this option.</p>
<i>Local working directory</i>	The path to your local working directory.
<i>HDFS working directory</i>	<p>The path to your Hadoop Distributed File System (HDFS) directory. If you leave this blank, Data Services uses <code>/user/sapds_hivetmp</code> as the default.</p> <div><p>i Note</p><p>If you use Beeline CLI, enter the directory that was created by your administrator with permission 755 assigned to each directory in the path.</p></div>
<i>String size</i>	The size of the Hive STRING datatype. The default is 100.
<i>SSL enabled</i>	<p>Select Yes to use a Secure Socket Layer (SSL) connection to connect to the Hive server.</p> <div><p>i Note</p><p>If you use Kerberos or Kerberos keytab for authentication, set this option to No.</p></div>

Option	Description
SSL Trust Store	The name of the trust store that verifies credentials and stores certificates.
Trust Store Password	The password associated with the trust store.
Authentication	<p>Indicates the type of authentication you are using for the Hive connection:</p> <p>Kerberos: Enter your Kerberos password in the Username and Password option.</p> <p>Kerberos keytab: The generated keytab file. Enter the keytab file location in Kerberos Keytab Location option.</p> <p>A Kerberos keytab file contains a list of authorized users for a specific password. The software uses the keytab information instead of the entered password in the Username and Password option. For more information about keytabs, see the MIT Kerberos documentation on the Massachusetts Institute of Technology (MIT) website.</p> <p>Data Services supports Kerberos authentication for Hadoop and Hive data sources when you use Hadoop and Hive services that are Kerberos enabled.</p> <div> <p>Note</p> <ul style="list-style-type: none"> You cannot use SSL and Kerberos or Kerberos keytab authentication together. Set the SSL enabled option to No when using Kerberos authentication. To enable SASL-QOP support for Kerberos, enter a <code>sasl.qop</code> value into the Additional Properties field. For more information, see the Additional Properties field description. </div> <p>To use Kerberos authentication, do the following:</p> <ol style="list-style-type: none"> 1. Install Kerberos 5 client 64-bit packages (<code>krb5</code>, <code>krb5-client</code>). 2. Configure Kerberos KDC according to the Hadoop Hive distribution requirements. 3. Make sure the Kerberos configuration file (<code>krb5.conf</code>) is available and contains the correct REALM or KDC configurations. Note that the location is installation-specific, under <code>/etc/krb5.conf</code> on Linux. 4. Point <code>/usr/lib64: linkrb5.so</code> to the preferred version of <code>libkrb5.so.<version></code> library. <p>For more information about Kerberos, see the Massachusetts Institute of Technology (MIT) website.</p>
Kerberos Realm	Specifies the name of your Kerberos realm. A realm contains the services host machines, application servers, and so on, that users can access. For example, <code>BIGDATA.COM</code> .

Option	Description
<i>Kerberos KDC</i>	<p>Specifies the server name of the Key Distribution Center (KDC). Secret keys for user machines and services are stored in the KDC database.</p> <p>Configure the Kerberos KDC with renewable tickets (ticket validity as required by Hadoop Hive installation).</p> <div> i Note Data Services supports MIT KDC and Microsoft AD for Kerberos authentication. </div>
<i>Kerberos Hive Principal</i>	The Hive principal name for the KDC. The name can be the same as the user name that you use when installing Data Services. Find the Hive service principal information in the <code>hive-site.xml</code> file. For example, <code>hive/<hostname>/@realm</code> .
<i>Kerberos Keytab Location</i>	<p>Location for the applicable Kerberos keytab that you generated for this connection.</p> <p>See the description for Authentication for more information about Kerberos keytab authentication.</p>
<i>Additional Properties</i>	<p>Specify any additional connection properties. Follow property value pairs with a semi-colon (;). Separate multiple property value pairs with a semicolon. For example:</p> <pre>name1=value1;</pre> <pre>name1=value1; name2=value2;</pre> <p>To enable SASL-QOP support, set the Authentication option to Kerberos. Then enter one of the following values, which should match the value on the Hive server:</p> <ul style="list-style-type: none"> Use <code>;sas1.qop=auth;</code> for authentication only. Use <code>;sas1.qop=auth-int;</code> for authentication with integrity protection. Use <code>;sas1.qop=auth-conf;</code> for authentication with integrity and confidentiality protection.

Parent topic: [Hive datastores \[page 78\]](#)

Related Information

[Using an ODBC driver to connect to a remote Hive Server \[page 82\]](#)

[Configuring bulk loading for Hive \[page 84\]](#)

5.3.7.3.5.2 Using an ODBC driver to connect to a remote Hive Server

Use a supported Hive ODBC driver to create a DSN (data source name) to remotely connect to your Hive server.

Before you create a Hive datastore, create a DSN using a supported Hive ODBC driver. Supported Hive ODBC drivers include the following:

- Cloudera
- Hortonworks
- MapR

Obtain driver downloads and related information by going to the product Web page for the driver type that you select.

In the following steps, we use the Cloudera ODBC driver for Hive as an example.

1. Download and install the Cloudera ODBC driver for Hive.
2. Configure a Data Source Name (DSN).

For Windows:

- a. Open the ODBC Data Source Administrator from your Windows Start menu. Or, use the ODBC administrator in the Datastore Editor when you create the Hive datastore in Data Services.
- b. In the ODBC Administrator, open the *System DSN* tab and select the Cloudera ODBC driver for Hive that you just installed.
- c. Click *Configure*.
- d. Enter the required information into the *Cloudera ODBC Driver for Hive DSN Setup* window and click *Save*.

Required information is based on the type of ODBC driver you select.

For Linux:

- a. Run the Connection Manager utility: `DSConnectionManager.sh`.
- b. Type the number that corresponds to the database type.

For example, type *15* for Hive Server:

```
Specify one database type:
1) MySQL
2) Microsoft SQL Server via DataDirect
3) SAP HANA
4) IBM DB2 oniSeries or zSeries via DataDirect
5) Teradata
6) Netezza NPS
7) Sybase IQ
8) Sybase ASE
9) IBM Informix IDS
10) Attunity
11) SQL Anywhere
12) HP Vertica
13) Amazon Redshift
14) Apache Cassandra
15) Hive Server
Specify database index #'1'
15
```

- c. Complete the remaining options. The Connection Manager creates the DSN.

Find complete instructions for using the Connection Manager in the *Administrator Guide*.

3. After you have configured the Hive DSN with Connection Manager, check to see that the Connection Manager created the following configuration files:

- `odbc.ini`
- `ds_odbc.ini`

i Note

If you currently have other ODBC connections configured, check to see that the connection information is added to the `odbc.ini` and `ds_odbc.ini` configuration files.

The `odbc.ini` is located in `$ODBCINI`:

```
[hive_cdr_pwd_dsn]
Description=Cloudera ODBC Driver for Apache Hive
Driver=/opt/cloudera/hiveodbc/lib/64/libclouderahiveodbc64.so
HOST=vantgvm1nx077.dhcp.pgdev.sap.corp
PORT=10000
Schema=default
ServiceDiscoveryMode=0
HiveServerType=2
AuthMech=3
ThriftTransport=1
UID=I852017
PWD=Initial1
KrbHostFQDN=vantgvm1nx035
KrbServiceName=hive
KrbRealm=DSQA366.CORP
SSL=0
```

The `ds_odbc.ini` file is located in [<LINK_DIR>](#)/bin:

```
[hive_cdr_pwd_dsn]
Driver=/usr/local/unixODBC-2.3.2/lib/libodbc.so
```

4. Create a Hive database datastore using the DSN you just created.

Import tables from the Hive Server using the Hive database datastore. Use an imported table as a source or target in your data flow. Preview the data by clicking the magnifying glass icon in the lower right corner of the source icon in your data flow. You can also use a Hive database datastore template table as the source or target.

Task overview: [Hive datastores \[page 78\]](#)

Related Information

[Hive adapter datastore configuration options \[page 79\]](#)

[Configuring bulk loading for Hive \[page 84\]](#)

[Datastore editor \[page 57\]](#)

5.3.7.3.5.3 Configuring bulk loading for Hive

Use a combination of Hadoop objects to configure bulk loading for Hive targets in a data flow.

Create the following objects:

- HDFS file location object
- HDFS file format
- Hive database datastore

To set up bulk loading to Hive, follow these steps:

1. Open the [Format](#) tab in the Local Object Library and expand the [HDFS Files](#) node.
2. Select the HDFS file format that you created for this task and drag it onto your data flow workspace.
3. Select [Make Source](#).
4. Add the applicable transform objects to your data flow.
5. Add a template table as a target to the data flow:
 - a. Select the template table icon from the tool palette at right.
 - b. Click on a blank space in your data flow workspace

The [Create Template](#) dialog box opens

6. Complete [Template name](#) with a new name for the target.
7. Select the Hive database datastore that you created for this task from the [In datastore](#) dropdown list.
8. Select a format from the [Formats](#) dropdown list.
9. Click [OK](#).
10. Connect the template to the data flow.
11. In your data flow workspace, open the target table and open the [Bulk Loader Options](#) tab.

The [Bulk Load](#) option is selected by default.

12. Select a mode from the [Mode](#) dropdown list.

Because the target is a newly-created table, there is no data in the table. However, if you use the data flow in subsequent runs, the [Mode](#) affects the data in the target table.

- [Append](#): Adds new records generated from Data Services processing to the existing data in the target table.
 - [Truncate](#): Replaces all existing records in the existing target table with the records generated from Data Services processing.
13. Select the HDFS file location object that you created for this task from the [HDFS File Location](#) dropdown list.
 14. Complete the remaining target options as applicable.

Task overview: [Hive datastores \[page 78\]](#)

Related Information

[Hive adapter datastore configuration options \[page 79\]](#)

[Using an ODBC driver to connect to a remote Hive Server \[page 82\]](#)

[Hadoop overview \[page 1427\]](#)

5.3.7.3.6 Attunity Connector

Create an Attunity Connector database datastore for use with main frame systems only.

Main window

Attunity option	Description
<i>CDC options</i>	<i>No CDC</i> : Select to not use Change Data Capture. <i>Native CDC</i> : Select to use Attunity Change Data Capture.
<i>Data source name</i>	Enter the Attunity data source name or names as defined in Attunity Studio. Separate multiple data source names with semicolons.
<i>Host location</i>	Enter the computer name, fully qualified domain name, or IP address of the Attunity server computer (host).
<i>Port</i>	Enter the port number for the Attunity server.
<i>Attunity workspace</i>	Enter the workspace name under which the data sources are defined in Attunity Studio. For information, refer to your database requirements.
<i>User name</i>	Enter the user name of the account through which SAP Data Services accesses the database.
<i>Password</i>	Enter the password associated with the user name that you enter.

Advanced options

Attunity option	Description
<i>General</i> (The <i>General</i> options do not appear when you select <i>Native CDC</i> .)	
<i>Rows per commit</i>	Enter the maximum number of rows loaded to a target table before saving the data. The default is 1000. This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.
<i>Overflow file directory</i>	Enter the location or browse to the location of overflow files written by target tables in this datastore. You can use a variable for this value.
<i>Locale</i>	
<i>Language</i>	For complete information, see Locales and Multi-byte Functionality [page 1378] .
<i>Code page</i>	
<i>Server code page</i>	
<i>Session</i>	
<i>Additional session parameters</i>	Additional session parameters specified as a valid SQL statement. Enter multiple SQL statements delimited by a semicolon.

Attunity option	Description
Aliases (Click here to create)	
Aliases	Click to open Create New Alias popup dialog. Enter the alias name and click OK . Enter the owner name in the Alias row.

Related Information

[Enable Automatic Data Transfer \[page 70\]](#)

[Change data capture \(CDC\) \[page 70\]](#)

5.3.7.3.7 Data Federator

Use the Data Federator database datastore to access data for read only.

Database datastore option descriptions

Main window options

Option	Description
Database Version	Select the Data Federator version from the dropdown list.
User Name	Enter the user name associated with the database.
Password	Enter the password associated with User Name .

Advanced option descriptions

Option	Description
General	
Rows per commit	<p>Enter the maximum number of rows loaded to a target table before saving the data. The default is 1000.</p> <p>This value is the default commit size for target tables in this datastore.</p> <p>You can overwrite this value for individual target tables.</p>
Bulk loader directory	<p>Enter the location where data files are written for bulk loading. You can enter a variable for this option.</p> <p>The default value is <code><DS_COMMON_DIR>/log/bulkload</code>.</p>

Option	Description
<i>Overflow file directory</i>	Enter the location of overflow files written by target tables in this datastore. You can use a variable for this option.
<i>Database server working directory</i>	Enter the working directory for the load utility on the computer that runs the Data Federator server. Complete this field whenever the Data Federator server and the Job Server run on separate machines.
<i>Locale</i>	
<i>Language</i>	For information, see Locales and Multi-byte Functionality [page 1378]
<i>Code page</i>	
<i>Server code page</i>	
<i>FTP</i>	
<i>FTP hostname</i>	<p>Enter the computer name, fully qualified domain name, or IP address.</p> <p>If this field is left blank or contains the name of the computer (host) where the Job Server resides, the software assumes that Data Federator and the software share the same computer and that FTP is unnecessary. When FTP is unnecessary, all other FTP-related fields can remain blank.</p>
<i>FTP login user name</i>	Enter the FTP user name. Required to use FTP.
<i>FTP login password</i>	Enter the password related to the FTP logon user name. Required to use FTP.
<i>FTP host working directory</i>	<p>Enter the absolute file path to the location on the database server where Data Services transfers the data file between the Job Server and the Data Federator server.</p> <p>For Windows servers, enter the virtual file path</p>
<i>ODBC Miscellaneous</i>	
<i>Date format</i>	<p>Enter a date format that Data Federator and the ODBC driver supports.</p> <p>The default date format in Data Services is <code>yyyy-mm-dd</code>.</p>
<i>Time format</i>	<p>Enter a time format that Data Federator and the ODBC driver supports.</p> <p>The default time format in Data Services is <code>hh:mm:ss</code>.</p>

Option	Description
Date-time format	<p>Enter a date/time format that Data Federator and the ODBC driver supports.</p> <p>The default date/time format in Data Services is <code>yyyy-mm-dd hh:mi:ss</code>.</p>
Decimal separator	<p>Enter the character that Data Federator uses to separate the decimal portion of a number. The default decimal separator for Data Services is a period.</p>
Data type conversion support	<p>Select an explicit convert function call that the software generates when there is a data type mismatch in an expression. The default is Automatic. Values include:</p> <ul style="list-style-type: none"> • Automatic • ODBC syntax • SQL-92 syntax • No: Disables this option. <p>For descriptions of values, see ODBC [page 108]</p>
NVL support	<p>Select the value to use when the input value is NULL. The default is Automatic. Values include:</p> <ul style="list-style-type: none"> • Automatic • ODBC syntax • No: Disables this option. <p>For descriptions of values, see ODBC [page 108]</p>
Ifthenelse support	<p>Yes: Allows conditional logic in mapping and selection operations.</p> <p>No: Does not allow conditional logic in mapping and selection operations.</p>
Session	
Additional session parameters	<p>Enter additional session parameters specified as a valid SQL statement. Delimit multiple SQL statements with semicolons.</p>
Aliases (Click here to create)	
Aliases	<p>Click the option to open the Create New Alias popup dialog. Enter an alias name and click OK. Then enter the alias owner name in the Alias row.</p>

5.3.7.3.8 DB2

Create a DB2 database datastore to use as a source or target in Data Services.

Main window

Option	Description
Database version	Select the version of your DB2 client. This is the version of DB2 that this data-store accesses.
CDC options	<p>Replication Server CDC: Uses the Replication Server to use change data capture.</p> <p>No CDC: Disables CDC processing. This is the default setting.</p>
Use data source name (DSN)	<p>Select to use data source name (DSN) to connect to the database.</p> <p>When you select DSN, also complete the following option:</p> <ul style="list-style-type: none">• Data Source Name <p>By default, this option is not selected and the software uses a server name (also known as DSN-less) connection. When you do not select DSN, also complete the following options:</p> <ul style="list-style-type: none">• Database server name• Database name• Port
Database server name	<p>Type the DB2 database server name.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Database name	<p>Type the name of the database defined in DB2.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Port	<p>Enter the number of the database port.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Data source name	<p>Type the data source name defined in DB2 for connecting to your database.</p> <p>This option is required when you select Use data source name (DSN).</p> <p>If you use the auto correct load feature for DB2 targets, ensure that your data source allows your user name to create or replace stored procedures.</p>
User name	Enter the user name of the account through which SAP Data Services accesses the database.
Password	Enter the password associated with the user.
Enable Automatic Data Transfer	For information, see Enable Automatic Data Transfer [page 70] .

General

Option	Description
<i>Rows per commit</i>	<p>Enter the maximum number of rows loaded to a target table before saving the data. The default is 1000.</p> <p>This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.</p>
<i>Bulk loader directory</i>	<p>Enter the location where command and data files are written for bulk loading. For Solaris systems, the path name must be less than 80 characters.</p> <p>You can enter a variable for this option.</p>
<i>Overflow file directory</i>	Enter the location of overflow files written by target tables in this datastore. A variable can also be used.

Locale

DB2 option	Description
<i>Language</i>	For more information, see Locales and Multi-byte Functionality [page 1378] .
<i>Code page</i>	
<i>Server code page</i>	

Bulk loader

Option	Description
<i>Bulk loader user name</i>	The name used when loading data with the bulk loader option. For bulk loading, you might specify a different user name. For example, specify a user who has import and load permissions.
<i>Bulk loader password</i>	The password used when loading with the bulk loader option.
<i>DB2 server working directory</i>	The working directory for the load utility on the computer that runs the DB2 server. You must complete this field whenever the DB2 server and the Job Server run on separate machines.

FTP

Option	Description
<i>FTP host name</i>	<p>Enter the Computer name, fully qualified domain name, or IP address of the FTP host.</p> <p>If you leave this field blank or it contains the name of the computer (host) where the Job Server resides, the software assumes that DB2 and the software share the same computer and that FTP is unnecessary.</p> <p>When FTP is unnecessary, you can leave all other FTP-related fields blank.</p>
<i>FTP login user name</i>	Must be defined to use FTP.
<i>FTP login password</i>	Must be defined to use FTP.
<i>Session</i>	
<i>Additional session parameters</i>	Additional session parameters specified as a valid SQL statement. Delimit multiple SQL statements with semicolons.

Option	Description
Aliases (Click here to create)	
Aliases	Enter the alias name and the owner name to which the alias name maps.
Linked Datastores (Click here to create) group	
Datastore name	The name of a datastore to which you linked the current datastore configuration in preparation to import a database link

Related Information

[Enable Automatic Data Transfer \[page 70\]](#)

[Change data capture \(CDC\) \[page 70\]](#)

5.3.7.3.9 HP Vertica datastore options

Create an HP Vertica database datastore to use as a source or target in a data flow.

After you create the HP Vertica database datastore, you can import HP Vertica tables into Data Services. Use the tables as source or targets in a dataflow, and create HP Vertica template tables.

Configure an HP Vertica database datastore using a supported driver and a DSN (data source name). See the *Product Availability Matrix (PAM)* for driver information.

SSL protocol is available for HP Vertica database datastores. Before you can create an SSL-enabled HP Vertica datastore, the HP Vertica database administrator user must install and configure MIT Kerberos 5 and create a DSN in the ODBC Data Source Administrator (Windows) or with DSCONNECTION Manager (Unix). For more information, see the *Designer Guide*.

Main window

HP Vertica option	Description
Database version	Select your HP Vertica client version from the drop-down list. This is the version of HP Vertica that this datastore accesses.
Data source name	<p>Required. Select a DSN from the dropdown list if you have already defined one. If you haven't defined a DSN previously, click ODBC Admin to define a DSN.</p> <p>You must first install and configure MIT Kerberos 5 and perform other HP Vertica set up tasks before you can define a DSN.</p> <p>For more information about HP Vertica MIT Kerberos and DSN for HP Vertica, read the Server Management section of the <i>Administrator Guide</i>.</p>

HP Vertica option	Description
User name	Enter the user name of the account through which SAP Data Services accesses the database.
Password	Enter the database password for the user that you entered in User Name .
Connection	
HP Vertica option	Description
Additional connection parameters	Enter information for any additional connection parameters. Use the format: <code><parameter1=value1; parameter2=value2></code>
General	
HP Vertica option	Description
Rows per commit	<p>Enter the maximum number of rows loaded to a target table before saving the data. The default is 1000.</p> <p>This value is the default commit size for target tables in this datastore.</p> <p>You can overwrite this value for individual target tables.</p>
Overflow file directory	Enter the location or browse to the location where the software writes overflow files written by target tables in this datastore. Must be defined to use FTP.
Session	
HP Vertica option	Description
Additional session parameters	Additional session parameters specified as valid SQL statements. Delimit multiple statements with a semicolon.
Aliases (Click here to create)	
HP Vertica option	Description
Aliases	Click the option to open a Create New Alias window. Save the alias and enter the owner name in the datastore editor.

Related Information

[Working with Aliases \[page 66\]](#)

[Enable Automatic Data Transfer \[page 70\]](#)

5.3.7.3.10 Informix

Create an Informix database datastore to use in Data Services.

Main window

Option	Description
Database version	Select the version of your Informix client. This version is the version of Informix that this datastore accesses.
Use data source name (DSN)	<p>Select to use DSN to connect to the database. When you select to use DSN, complete the Data source name option.</p> <p>By default, this option is not selected and the software uses a server name (also known as DSN-less) connection. For a DSN-less connection, complete the following options:</p> <ul style="list-style-type: none">• Database server name• Database name• Port
Database server name	<p>Type the Informix database server name. This name is the Informix server instance name, not the host name.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Database name	<p>Type the name of the database defined in Informix .</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Port	<p>Type the port number to connect to this database.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Data source name	<p>Select the DSN from the dropdown list.</p> <p>This option is required when you select Use data source name (DSN). Create a DSN by clicking ODBC Admin or by using the DSCConnection Manager (Unix). For information about creating a DSN with a supported driver, see the Designer Guide.</p>
User name	Enter the user name of the account through which SAP Data Services accesses the database.
Password	Enter the password associated with the User name .

General

Option	Description
Rows per commit	Enter the maximum number of rows loaded to a target table before saving the data. The default is 1000. This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.

Option	Description
Bulk loader directory	Enter the directory where the software writes SQL, control, commands, and data files for bulk loading.
<div> i Note For Solaris systems, the path name must be less than 80 characters. </div>	
	You can enter a variable for this option.
Overflow file directory	Enter the location of overflow files written by target tables in this datastore.

Locale

Option	Description
Language	For more information, see Locales and Multi-byte Functionality [page 1378] .
Code page	
Server code page	

Session

Option	Description
Additional session parameters	Additional session parameters specified as a valid SQL statement. Delimit with a semicolon for multiple SQL statements.
Aliases (Click here to create)	
Aliases	Click to open the Create New Alias popup dialog. Enter an alias name, click OK , and enter the owner name in the Alias text row.

i Note

To use large object types with Informix, enable large object support when you create the DSN with the Informix driver.

- For Windows, open the ODBC Administrator, select the Informix driver, and click [Configure](#). The [Informix ODBC Driver Setup](#) dialog box opens. Select [Report Standard ODBC Types only](#) to enable large object support.
- For Unix, edit the `odbc.ini` file to set the option `NEEDODBCTYPESONLY=1`.

For more information about Informix ODBC driver configuration, see your Informix ODBC driver configuration documentation.

Related Information

[Enable Automatic Data Transfer \[page 70\]](#)

5.3.7.3.11 Memory

Create a Memory database datastore for use in Data Services.

Bulk Loader

Option	Description
JS and DB on same machine	<p>Yes: The Job Server and the database are on the same computer.</p> <p>No: The Job Server and the database are on separate computers. This is the default setting.</p>

Related Information

[Locales and Multi-byte Functionality \[page 1378\]](#)

5.3.7.3.12 Microsoft SQL Server

Create a Microsoft SQL Server database datastore for Data Services.

Note

DataDirect error messages for ODBC driver. If there is a problem when processing Microsoft SQL Server data using a UNIX Job Server, the software issues an error message that might contain a DataDirect internal error message. DataDirect error-message syntax resembles the following structure: `<vendor-ODBCcomponent: message>`. See the documentation on the DataDirect Web site for more information about DataDirect message text or codes. See the *Administrator Guide* for information about configuring DataDirect ODBC for Data Services.

Main window

Option	Description
CDC options	<p>Options include the following:</p> <ul style="list-style-type: none">• No CDC: Disables change data capture.• Native CDC: Uses Microsoft SQL Server change data capture.• Replication Server CDC: Uses the Replication Server change data capture. <p>If you choose Replication Server CDC and you enable Use SSL encryption in the Advanced options, the SSL encryption applies only to the datastore connection and not the replication server connection.</p>

Option	Description
<i>Database subtype</i>	<p>Select the database subtype. Options include:</p> <ul style="list-style-type: none"> • <i>Azure Paas</i> • <i>Azure VM</i> • <i>On Premise</i> • <i>Azure DW</i> • <i>APS</i> <p><i>On Premise</i> is required when you enable SSL encryption.</p> <div> <p>i Note</p> <p>In Azure DW and APS, you cannot use a create table, truncate table, or drop table statement in a preload or postload statement.</p> </div>
<i>Database version</i>	Select the version of your SQL Server client that this datastore accesses.
<i>Database server name</i>	Enter the computer name, fully qualified domain name, or IP address of the machine where the SQL Server instance is located.
<i>Database name</i>	Enter the name of the database to which the datastore connects.
<i>User name</i>	Enter the user name of the account through which Data Services accesses the database.
<i>Password</i>	Enter the password associated with the user in <i>User name</i> .
<i>Connection</i>	
<i>Use Windows Authentication</i>	<p><i>No</i>: Uses Microsoft SQL Server authentication to connect to this datastore. This is the default setting.</p> <p><i>Yes</i>: Uses Windows authentication to connect to this datastore.</p> <p>For more information on how to use Windows authentication with Microsoft SQL Server, refer to the Microsoft SQL Server documentation.</p> <div> <p>i Note</p> <p>Windows authentication is not supported on UNIX.</p> </div>

Option	Description
Use SSL encryption	<p>This option is only available when you choose On Premise for the Database Sub-Type.</p> <p>No: Does not use SSL encryption. This is the default setting.</p> <p>Yes: Uses SSL encryption for data that is transmitted across a network from the Microsoft SQL Server database to SAP Data Services.</p> <p>If you choose the Replication Server CDC option and SSL encryption, SSL encryption does not apply to the replication server connection, it only applies to the data-store connection.</p> <div> <p>Note</p> <p>This option provides SSL encryption for datastore connections. This option does not affect Replication Server CDC connections.</p> </div> <p>Selecting this option slows down job performance.</p> <p>For SSL encryption to work, your administrator must perform prerequisite steps to configure the cryptographic library.</p> <p>For more information about administrator tasks for enabling SSL encryption, see the <i>Administrator Guide</i>.</p>

General

Option	Description
Rows per commit	Enter the maximum number of rows loaded to a target table before saving the data. The default is 1000. This value is the default commit size for target tables in this data-store. You can overwrite this value for individual target tables.
Overflow file directory	Enter the location of or browse to the location of the overflow files written by target tables in this datastore. You can enter a variable for this option.

Locale

Option	Description
Language	For more information, see Locales and Multi-byte Functionality [page 1378] .
Code page	
Server code page	

Session

Option	Description
Additional session parameters	Additional session parameters specified as a valid SQL statement. Delimit multiple SQL statements with a semicolon.

Aliases (Click here to create)

Option	Description
Aliases	Click to open the Create New Alias popup dialog. Enter an alias name, click OK , and enter the owner name in the Alias text row.

Linked Datastores (Click here to create)

Option	Description
Datastore Name	The name of a datastore to which you linked the current datastore configuration in preparation to import a database link

Related Information

[Enable Automatic Data Transfer \[page 70\]](#)

[Change data capture \(CDC\) \[page 70\]](#)

5.3.7.3.13 MySQL

Create a MySQL database datastore for Data Services.

Main window

Option	Description
Database version	Select the version of the MySQL client that this datastore accesses.
Use data source name (DSN)	<p>Select to use DSN to connect to the database. When selected, also complete the Data source name option.</p> <p>The default setting is not selected, and the software uses a server name (also known as DSN-less) connection. For a DSN-less connection, complete the following additional options:</p> <ul style="list-style-type: none">• Database server name• Database name• Port
Database server name	<p>Enter the MySQL database server name.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Database name	<p>Enter the name of the database defined in MySQL.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Port	<p>Enter the number of the database port.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Data source name	<p>Enter the data source name defined in MySQL for connecting to your database.</p> <p>This option is required when you select Use data source name (DSN).</p> <p>If you use the Auto correct load feature for MySQL targets, ensure that your data source allows your user name to create or replace stored procedures.</p>
User name	Enter the user name of the account through which SAP Data Services accesses the database.

Option	Description
Password	Enter the user password.

Connection

Option	Description
Additional connection parameters	<p>Enter information for any additional parameters that the data source ODBC driver and database supports. Use the following format:</p> <pre><parameter1=value1; parameter2=value2></pre>

General

Option	Description
Rows per commit	<p>Enter the maximum number of rows loaded to a target table before saving the data. The default is 1000.</p> <p>This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.</p>
Bulk loader directory	<p>Enter or browse to the location where command and data files are written for bulk loading. For Solaris systems, the path name must be less than 80 characters.</p> <p>You can enter a variable for this option.</p>
Overflow file directory	Enter or browse to the location of overflow files written by target tables in this datastore. A variable can also be used.
Database Server working directory	Enter or browse to the location of the working directory for the load utility on the computer that runs the MySQL server. Complete this option whenever the MySQL server and the Job Server run on separate machines.

Locale

Option	Description
Language	For more information, see Locales and Multi-byte Functionality [page 1378] .
Code page	
Server code page	

FTP

Option	Description
FTP host name	<p>Enter the FTP host name using the computer name, fully qualified domain name, or IP address.</p> <p>If you leave this field blank or you enter the name of the computer (host) where the Job Server resides, the software assumes that MySQL and the software share the same computer and that FTP is unnecessary.</p> <p>When FTP is unnecessary, all other FTP-related fields can remain blank.</p>
FTP login user name	Must be defined to use FTP.
FTP login password	Must be defined to use FTP.

Option	Description
FTP host working directory	Enter the absolute file path to the location on the database server where Data Services transfers the data file between the Job Server and the MySQL server. For Windows servers only, you can configure a path to a virtual directory.

ODBC Miscellaneous

Option	Description
Date format	Enter a date format supported by the data source (a date format that the data source ODBC driver and database supports). The default date format is <code>yyyy-mm-dd</code> .
Time format	Enter a time format supported by the data source (a time format that the data source ODBC driver and database supports). The default time format is <code>hh:mi:ss</code> .
Date-time format	Enter a date/time format supported by the data source (a date/time format supported by the data source ODBC driver and database). The default date/time format is <code>yyyy-mm-dd hh:mi:ss</code> .
Decimal separator	Enter the character that the data source uses to separate the decimal portion of a number. The default is a period.
Data type conversion support	When there's a data type mismatch in an expression, the software automatically generates an explicit convert function call. Options are: <ul style="list-style-type: none"> • Automatic (default) • ODBC syntax • SQL-92 syntax • No (disables this feature)
NVL support	If the input value is NULL, replace with the specified value. Options are: <ul style="list-style-type: none"> • Automatic (default) • ODBC syntax • No (disables this feature)
Ifthenelse support	No : Does not allow conditional logic in mapping and selection operations. This is the default. Yes : Allows conditional logic in mapping and selection operations.

Session

Option	Description
Additional session parameters	Additional session parameters specified as a valid SQL statement. Delimit multiple SQL statements with a semicolon.

Aliases (Click here to create)

Option	Description
Aliases	Enter the alias name and the owner name to which the alias name maps.

Related Information

[Enable Automatic Data Transfer \[page 70\]](#)

5.3.7.3.14 Netezza

Create a Netezza database datastore for Data Services.

Main window

Option	Description
Database version	Select the version of your Netezza client. This is the version of Netezza that this datastore accesses.
Use data source name (DSN)	<p>Select to use DSN to connect to the database. When you select to use DSN, complete the Data source name option.</p> <p>By default, this option is not selected and the software uses a server name (also known as DSN-less) connection. For a DSN-less connection, complete the following options:</p> <ul style="list-style-type: none">• Database server name• Database name• Port
Database server name	<p>Enter the Netezza database server name.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Database name	<p>Enter the name of the database defined in Netezza.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Port	<p>Enter the number of the database port.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Data source name	<p>Enter the data source name defined in Netezza for connecting to your database. This option is required when you select Use data source name (DSN).</p> <p>Create a DSN by clicking ODBC Admin or by using the DSCONNECTION Manager (Unix). For information about creating a DSN with a supported driver, see the <i>Designer Guide</i>.</p> <p>If you use the Auto correct load feature for Netezza targets, be sure that your data source allows your user name to create or replace stored procedures.</p>
User name	Enter the user name of the account through which SAP Data Services accesses the database.
Password	Enter the user password associated with the User name .
Enable automatic data transfer	<p>The Enable automatic data transfer checkbox is selected by default when you create a new datastore and you chose Database for Datastore type.</p> <p>Keep Enable automatic data transfer selected to enable transfer tables in this datastore that the Data_Transfer transform can use to push down subsequent database operations.</p>

Connection

Option	Description
Additional connection parameters	<p>Enter information for any additional parameters that the data source ODBC driver and database supports. Use the following format:</p> <pre><parameter1=value1; parameter2=value2></pre>

General

Option	Description
Rows per commit	<p>Enter the maximum number of rows loaded to a target table before saving the data. The default is 1000.</p> <p>This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.</p>
Bulk loader directory	<p>Enter or browse to the location where command and data files are written for bulk loading. For Solaris systems, the path name must be less than 80 characters.</p> <p>You can enter a variable for this option.</p>
Overflow file directory	Enter or browse to the location of overflow files written by target tables in this datastore. A variable can also be used.
Database Server working directory	Enter or browse to the location of the working directory for the load utility on the computer that runs the Netezza server. Complete this option whenever the Netezza server and the Job Server run on separate machines.

Locale

Option	Description
Language	For more information, see Locales and Multi-byte Functionality [page 1378] .
Code page	
Server code page	

FTP

Option	Description
FTP host name	<p>Enter the FTP host name using the computer name, fully qualified domain name, or IP address.</p> <p>If you leave this field blank or you enter the name of the computer (host) where the Job Server resides, the software assumes that MySQL and the software share the same computer and that FTP is unnecessary.</p> <p>When FTP is unnecessary, all other FTP-related fields can remain blank.</p>
FTP login user name	Must be defined to use FTP.
FTP login password	Must be defined to use FTP.
FTP host working directory	<p>Enter the absolute file path to the location on the database server where Data Services transfers the data file between the Job Server and the MySQL server.</p> <p>For Windows servers only, you can configure a path to a virtual directory.</p>

ODBC Miscellaneous

Option	Description
Date format	Enter a date format supported by the data source (a date format that the data source ODBC driver and database supports). The default date format is <code>yyyy-mm-dd</code> .
Time format	Enter a time format supported by the data source (a time format that the data source ODBC driver and database supports). The default time format is <code>hh:mi:ss</code> .
Date-time format	Enter a date/time format supported by the data source (a date/time format supported by the data source ODBC driver and database). The default date/time format is <code>yyyy-mm-dd hh:mi:ss</code> .
Decimal separator	Enter the character that the data source uses to separate the decimal portion of a number. The default is a period.
Data type conversion support	<p>When there's a data type mismatch in an expression, the software automatically generates an explicit convert function call. Options are:</p> <ul style="list-style-type: none"> • Automatic (default) • ODBC syntax • SQL-92 syntax • No <p>See ODBC [page 108] for value descriptions.</p>
NVL support	<p>If the input value is NULL, replace with the specified value. Options are:</p> <ul style="list-style-type: none"> • Automatic (default) • ODBC syntax • No (disables this feature) <p>See ODBC [page 108] for value descriptions.</p>
Ifthenelse support	<p>No: Does not allow conditional logic in mapping and selection operations. This is the default.</p> <p>Yes: Allows conditional logic in mapping and selection operations.</p>

Session

Option	Description
Additional session parameters	Additional session parameters specified as a valid SQL statement. Delimit multiple SQL statements with a semicolon.

Aliases (Click here to create)

Option	Description
Aliases	Click to open the Create New Alias popup dialog. Enter an alias name, click OK , and enter the owner name in the Alias text row.

Related Information

[Enable Automatic Data Transfer \[page 70\]](#)

5.3.7.3.15 ODBC editor fields and options

Create an ODBC database datastore to use in Data Services.

Main window

Option	Description
Data source name	Select the Data Source Name (DSN) from the dropdown list.
User name	Enter the user name of the account through which the software accesses the database.
Password	Enter the user password associated with User name .
ODBC Admin button	Click to launch the Windows ODBC Data Source Administrator. Select a supported driver for the database and configure a data source. After closing the ODBC Data Source Administrator, select the newly created data source name from the dropdown list. For Unix, use the DSCConnection Manager to create a DSN.

Connection

Option	Description
Additional connection information	Enter information for any additional connection parameters. Ensure that the parameters are supported by the ODBC driver and the database. Use the format: <parameter1=value1; parameter2=value2>

General

Option	Description
Rows per commit	Enter the maximum number of rows loaded to a target table before saving the data. The default is 1000. This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.
Bulk loader directory	Enter the directory where the software writes SQL, control, command, and data files for bulk loading. <div>Note For Solaris systems, the path name must be less than 80 characters.</div> You can enter a variable for this option.
Overflow file directory	Enter the location for overflow files written by target tables in this datastore. You can enter a variable for this option.
Database server working directory	Enter or browse for the working directory on the database server that stores files such as logs. Must be defined to use FTP.

Locale

Option	Description
Language	For more information, see Locales and Multi-byte Functionality [page 1378] .
Code page	
Server code page	

FTP

i Note

If this datastore is not being used specifically for Netezza bulk loading, the software ignores any FTP option entries.

Option	Description
<i>FTP host name</i>	For a Netezza server, type the name of the Netezza server computer (host). Must be defined to use FTP. Use the computer name, fully qualified domain name, or IP address.
<i>FTP login user name</i>	Must be defined to use FTP.
<i>FTP login password</i>	Must be defined to use FTP.
<i>FTP host working directory</i>	The location on the database server from where the software retrieves diagnostic files generated by the database's bulk loader. It must be accessible from the FTP server. It is usually the same as the database's working directory. If unsure, contact your system administrator.

i Note

Configure the FTP server to accept an absolute path.

The options in the following table have one or more of the following values:

- Automatic
- ODBC syntax
- SQL-92 syntax
- Yes
- No

The default setting is *Automatic*. For descriptions of values in the following table, see [ODBC \[page 108\]](#).

ODBC Capability Support

Option	Description
<i>Array fetch</i>	Select <i>No</i> if you encounter errors when reading from an ODBC datastore. For example, if you receive an error message that involves the ODBC call <code>SQLFetchScroll</code> , assume that your ODBC driver does not support array fetch. Selecting <i>No</i> turns off the array fetch capability. With the <i>Array fetch</i> option disabled, the software fetches one row at a time from the ODBC data source. The software does not display the <i>Array fetch size</i> performance option for any source table editors and SQL transform editors that use the ODBC datastore.
<i>Parameterized SQL</i>	The software generates SQL statements with parameters instead of literal values, which can significantly improve performance.
<i>Outer join</i>	Determines whether the ODBC driver supports outer join syntax.
<i>Auto commit</i>	Determines whether the ODBC driver supports auto commit.
<i>Server sorts in binary</i>	Determines whether the server performs binary sort.

For descriptions of the values for the following options, see [ODBC \[page 108\]](#).

ODBC Math Function Support

Option	Description
<i>Absolute</i>	Returns the absolute value of an input number.
<i>Ceiling</i>	Returns the smallest integer value greater than or equal to an input number.
<i>Floor</i>	Returns the largest integer value less than or equal to an input number.
<i>Round</i>	Rounds a given number to the specified precision.
<i>Truncate</i>	Truncates a given number to the specified precision.
<i>Sqrt</i>	Returns the square root of the input number.
<i>Log</i>	Returns the base-10 logarithm of the given numeric expression.
<i>Ln</i>	Returns the natural logarithm of the given numeric expression.
<i>Power</i>	Returns the value of the given expression to the specified power.
<i>Mod</i>	Returns the remainder when one number is divided by another.

For descriptions of the values for the following options, see [ODBC \[page 108\]](#).

ODBC String Function Support

Option	Description
<i>Lower case</i>	Changes the characters in a string to lowercase.
<i>Upper case</i>	Changes the characters in a string to uppercase.
<i>Rtrim blanks</i>	Removes blank characters from the end of a string.
<i>Ltrim blanks</i>	Removes blank characters from the start of a string.
<i>Length</i>	Returns the number of characters in a given string.
<i>Substring</i>	Returns a specific portion of a string starting at a given point in the string.
<i>Soundex</i>	Returns the soundex encoding of the input string.

For descriptions of the values for the following options, see [ODBC \[page 108\]](#).

ODBC Date Function Support

Option	Description
<i>System date</i>	Select the method to use to return the current date as listed by the Job Server operating system.
<i>System time</i>	Select the method to use to return the current time as listed by the operating system.
<i>Week</i>	Select the method to use to determine the week in the year in which the given date falls.
<i>Month</i>	Select the method to use to determine the month in which the given date falls.
<i>Quarter</i>	Select the method to use to determine the quarter in which the given date falls.
<i>Year</i>	Select the method to use to determine the year in which the given date falls.
<i>Day of month</i>	Select the method to use to determine the day in the month on which the given date falls.
<i>Day of year</i>	Select the method to use to determine the day in the year on which the given date falls.

For descriptions of the values for the following options, see [ODBC \[page 108\]](#).

ODBC Aggregate Function Support

Option	Description
<i>Average</i>	Calculates the average of a given set of values.
<i>Count</i>	Counts the number of values in a table column.
<i>Count Distinct</i>	Counts the number of distinct non-NULL values in a table column.
<i>Max</i>	Returns the maximum value from a list.
<i>Min</i>	Returns the minimum value from a list.
<i>Sum</i>	Calculates the sum of a given set of values.

Miscellaneous

Option	Description
<i>Date format</i>	Enter a date format supported by the data source (a date format that the data source ODBC driver and database supports). The default date format is <code>yyyy-mm-dd</code> .
<i>Time format</i>	Enter a time format supported by the data source (a time format that the data source ODBC driver and database supports). The default time format is <code>hh:mi:ss</code> .
<i>Date-time format</i>	Enter a date/time format supported by the data source (a date/time format supported by the data source ODBC driver and database). The default date/time format is <code>yyyy-mm-dd hh:mi:ss</code> .
<i>Decimal separator</i>	Enter the character that the data source uses to separate the decimal portion of a number. The default is a period.
<i>Data type conversion support</i>	When there's a data type mismatch in an expression, the software automatically generates an explicit convert function call.
<i>NVL support</i>	If the input value is NULL, replace with the specified value.
<i>Ifthenelse support</i>	<p>No: Does not allow conditional logic in mapping and selection operations. This is the default.</p> <p>Yes: Allows conditional logic in mapping and selection operations.</p>
<i>NVARCHAR type name</i>	<p>Select an option for loading multibyte data to template tables based on the database type:</p> <ul style="list-style-type: none"> • <Unknown>: Select when you do not know the type name. • NVARCHAR: Select for all supported databases except Oracle. • NVARCHAR2: Select for Oracle databases only. <p>In the template table target editor, setting the option Use NVARCHAR for VARCHAR columns in supported databases to Yes enables this data type conversion. See also Template table [page 274].</p>

Session

Option	Description
<i>Additional session parameters</i>	Additional session parameters specified as a valid SQL statement. Delimit multiple SQL statements with a semicolon.

Aliases

Option	Description
Aliases	Click to open the Create New Alias popup dialog. Enter an alias name, click OK , and enter the owner name in the Alias text row.

Related Information

[Enable Automatic Data Transfer \[page 70\]](#)

5.3.7.3.15.1 ODBC

The ODBC database datastore has specific ODBC options for which you select values from a dropdown list.

To define an ODBC datastore connection, define a data source name (DSN), a user name, a password if applicable, and optionally a set of advanced options.

Selecting an ODBC data source name

Select a data source name in one of the following ways in the [Data source name](#) option of the ODBC datastore editor:

- Select an existing DSN from the [Data Source Name](#) dropdown list.
- Type the name of an existing DSN.
- Click [ODBC Admin](#) to launch the Windows ODBC Data Source Administrator where you create or configure data sources. After closing the ODBC Data Source Administrator, select the newly created DSN from the [Data Source Name](#) dropdown list.

Defining ODBC datastore options

Click [Advanced](#) to open the advanced options for defining an ODBC datastore. For each applicable option, select a value from the dropdown list or type a custom value when allowed.

The following table contains descriptions for the ODBC datastore options that appear in the dropdown lists.

Value	Description
<i>Automatic</i>	<p>For most options, this is the default value. Select <i>Automatic</i> if you do not know if the ODBC driver supports the option. With <i>Automatic</i> selected, SAP Data Services queries the driver to determine its capabilities:</p> <ul style="list-style-type: none"> • If the driver supports that option, Data Services pushes down the operation to the ODBC database. • If the ODBC driver does not support that option, Data Services executes the operation internally. <p>To circumvent possible inconsistencies with the ODBC driver, specify an option other than <i>Automatic</i>. If you select anything other than <i>Automatic</i>, the software does not query the driver for that particular capability.</p>
<i>ODBC syntax</i>	<p>The software assumes that the ODBC driver supports the option function or capability and uses ODBC syntax.</p> <p>For example, for the ABSOLUTE function, the ODBC syntax is:</p> <pre>{fn abs (TAB1.COL1)}</pre>
<i>SQL-92</i>	<p>The software assumes that the ODBC driver supports the option function or capability and uses SQL-92 syntax.</p> <p>For example, when the software generates an explicit CONVERT function, the SQL-92 syntax is:</p> <pre>CAST (TAB1.VC_COL AS SQL_INTEGER)</pre>
<i>No</i>	<p>The software assumes that the ODBC driver does not support the option function or capability and executes it internally.</p>
Custom	<p>Many functions allow you to type in the specific function call to use for that option. The software assumes that the ODBC driver supports the function capability.</p> <div> <p>i Note</p> <p>You cannot specify the signature of the function; it is the same as in the ODBC signature.</p> </div> <p>For example, for the string function <i>Upper case</i>, instead of using {fn ucase (. . .) }, type in the <i>Upper case</i> option field upper. The software generates the following:</p> <pre>upper (TAB1.VC_COL)</pre>

5.3.7.3.16 Oracle

Create an Oracle database datastore to use in Data Services.

The following tables contain descriptions for Oracle datastore options.

Main window

Option	Description
<i>Database version</i>	Select the version of your Oracle client. This is the version of Oracle that this datastore accesses.
<i>CDC options</i>	Options include the following: <ul style="list-style-type: none"> • <i>No CDC</i>: Disables change data capture. • <i>Native CDC</i>: Uses Microsoft SQL Server change data capture. • <i>Replication Server CDC</i>: Uses the Replication Server change data capture.
<i>Use TNS name</i>	Select to use TNS to connect to the database. By default, this option is not selected and the software uses a server name (also known as TNS-less) connection. For a TNS-less connection, complete the following options: <ul style="list-style-type: none"> • <i>Hostname</i> • <i>SID</i> • <i>Port</i> If you select the <i>Use TNS name</i> checkbox, also complete <i>TNS name</i>
<i>Hostname</i>	Enter the computer name, fully qualified domain name, or IP address of the machine where the Oracle Server instance is located. This option is required if you did not select <i>Use TNS name</i> .
<i>SID</i>	Enter the System ID for the Oracle database. This option is required if you did not select <i>Use TNS name</i> .
<i>Port</i>	Enter the port number to connect to this Oracle Server. This option is required if you did not select <i>Use TNS name</i> .
<i>TNS name</i>	Enter an existing Oracle Transparent Network Substrate (TNS) name through which the software accesses sources and targets defined in this datastore. This option is required when you select <i>Use TNS name</i> .
<i>User name</i>	Enter the user name of the account through which the software accesses the database.
<i>Password</i>	Enter the password associated with the <i>User name</i> .

General

Option	Description
<i>Rows per commit</i>	Enter the maximum number of rows loaded to a target table before saving the data. The default is 1000. This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.
<i>Bulk loader directory</i>	Enter the location where the software writes command and data files for bulk loading.

i Note

For Solaris systems, the path name must be less than 80 characters. You can enter a variable for this option.

Option	Description
Overflow file directory	Enter the location of overflow files written by target tables in this datastore. You can enter a variable for this option.

Locale

Option	Description
Language	For more information, see Locales and Multi-byte Functionality [page 1378] .
Code page	
Server code page	

Session

Option	Description
Additional session parameters	Additional session parameters specified as a valid SQL statement. Delimit multiple SQL statements with a semicolon.

Oracle Miscellaneous

Option	Description
Default precision for number	Enter the total number of digits in the value. Enter a value from 1 to 96.
Default scale for number	Enter the number of digits to the right of the decimal point. For example, 0 <= scale <= precision.

Aliases (Click here to create)

Option	Description
Aliases	Click to open the Create New Alias popup dialog. Enter an alias name, click <i>OK</i> , and enter the owner name in the Alias text row.

Linked Datastores (Click here to create)

Option	Description
Datastore Name	The name of a datastore to which you linked the current datastore configuration in preparation to import a database link

Related Information

[Enable Automatic Data Transfer \[page 70\]](#)

[Change data capture \(CDC\) \[page 70\]](#)

5.3.7.3.17 Persistent Cache

Option descriptions for setting up a persistent cache datastore.

Basic options

Option	Description
Cache directory	Enter or browse to the location of your cache directory.
Force UTF-16 codepage	<p>Affects how a dataflow runs when the datastore is used in a dataflow.</p> <ul style="list-style-type: none">• Selected: Caches data in a Unicode multi-byte format for persistent cache database types only.• Not selected: Runs the dataflow based on the engine runtime locale. <p>You may benefit by selecting this option when you have several data flows that are dependent on each other and you have a mix of single-and multi-byte data.</p> <p>Setting this option ensures that the dependent data flows are compatible. However, if you have single-byte data and you select this option, you may see performance degradation in the data flow that uses this persistent cache.</p>

5.3.7.3.18 SAP HANA datastore options

Create an SAP HANA database datastore to use in Data Services.

Beginning with SAP HANA 2.0 SPS 01 MDC, use a database datastore to access a specified tenant database.

Option	Description
Database version	Select the version of your SAP HANA database client (the version of the SAP HANA database that this datastore accesses).
Use data source name (DSN)	<p>Select to use a data source name (DSN) to connect to the database.</p> <div><p>i Note</p><p>For SSL encryption, use the DSN SSL that you created in “Configure DSN SSL for SAP HANA” in the <i>Administrator Guide</i>.</p></div> <p>This option is not selected by default. When not selected, the software uses a server name, also known as a DSN-less connection. For a DSN-less connection, complete the Database server name and Port options.</p>

Option	Description
Database server name	<p>Enter the name of the computer where the SAP HANA server is located.</p> <p>This option is required if you did not select Use data source name (DSN).</p> <p>If you are connecting to HANA MDC, enter the HANA database server name for the applicable tenant database.</p>
Port	<p>Enter the port number to connect to the SAP HANA Server. The default is 30015.</p> <p>This option is required if you did not select Use data source name (DSN).</p> <p>If you are connecting to SAP HANA 2.0 SPS 01 MDC or later, enter the port number of the specific tenant database.</p> <div> <p>i Note</p> <p>See SAP HANA documentation to learn how to find the specific tenant database port number.</p> </div>
Data source name	<p>Select or type the data source name that you defined in the ODBC Administrator for connecting to your database.</p> <div> <p>i Note</p> <p>For SSL encryption, use the DSN SSL that you created in "Configure DSN SSL for SAP HANA" in the <i>Administrator Guide</i>.</p> </div> <p>This option is required when you select Use data source name (DSN).</p>
User name	Enter the user name of the account through which the software accesses the database.
Password	Enter the password associated with the User name .
Enable automatic data transfer	<p>This option is selected by default under the following circumstance:</p> <ul style="list-style-type: none"> • When you create a new datastore and • When you choose Database for the option Datastore type. <p>Select this option so that the Data Transfer transform pushes down subsequent database operations to the database.</p>

Advanced options

Option	Description
Connection	
Database name	<p>Optional. Enter the specific tenant database name.</p> <p>Applicable for SAP HANA versions 2.0 SPS 01 MDC and later.</p>

Option	Description
<i>Additional connection information</i>	<p>Enter information for any additional parameters that the data source ODBC driver and database supports. Use the following format:</p> <pre><parameter1=value1; parameter2=value2></pre>
<i>General</i>	
<i>Rows per commit</i>	<p>Enter the maximum number of rows loaded to a target table before saving the data.</p> <p>This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.</p>
<i>Overflow file directory</i>	Enter the location of overflow files written by target tables in this datastore. You could also use a variable.
<i>Session</i>	
<i>Additional session parameters</i>	Additional session parameters specified as a valid SQL statement. Delimit multiple SQL statements with a semicolon.
<i>Aliases (Click here to create)</i>	
<i>Aliases</i>	Click to open the Create New Alias popup dialog. Enter an alias name, click <i>OK</i> , and enter the owner name in the Alias text row.

Related Information

[Enable Automatic Data Transfer \[page 70\]](#)

5.3.7.3.19 SAP SQL Anywhere

Create an SAP SQL Anywhere database datastore to use in Data Services.

Main window

Option	Description
<i>Database Version</i>	Select the applicable client version from the dropdown list for the SAP SQL Anywhere that you are using.

Option	Description
<i>Use data source name (DSN)</i>	<p>Select to use data source name (DSN) to connect to the database.</p> <p>When you select DSN, also complete the following option:</p> <ul style="list-style-type: none"> • <i>Data Source Name</i> <p>By default, this option is not selected and the software uses a server name (also known as DSN-less) connection. When you do not select DSN, also complete the following options:</p> <ul style="list-style-type: none"> • <i>Database server name</i> • <i>Database name</i> • <i>Port</i>
<i>Database Server Name</i>	<p>Enter the name of the computer where the SAP SQL Anywhere server is located.</p> <p>This option is required if you did not select <i>Use Data source name (DSN)</i>.</p>
<i>Database Name</i>	<p>Enter the database name.</p> <p>This option is required if you did not select <i>Use Data source name (DSN)</i>.</p>
<i>Port</i>	<p>Enter the port number to connect to the SAP SQL Anywhere Server.</p> <p>This option is required if you did not select <i>Use Data source name (DSN)</i>.</p>
<i>Data Source Name</i>	<p>Type the data source name defined in the database for connecting to your database.</p> <p>This option is required when you select <i>Use data source name (DSN)</i>.</p>
<i>User Name</i>	<p>Enter the applicable user name to access the database.</p>
<i>Password</i>	<p>Enter the password associated with the <i>User Name</i>.</p>
<i>Enable Automatic Data Transfer</i>	<p>Enables transfer tables in this datastore, which the Data_Transfer transform can use to push down subsequent database operations.</p> <p>This option is enabled by default.</p>

Connection

Option	Description
Additional connection parameters	Enter information for any additional connection parameters. Use the format <code><parameter1>=<value1>;<parameter2>=<value2></code>

General

Option	Description
Rows per commit	Enter the maximum number of rows loaded to a target table before saving the data. The default is 1000. This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.
Overflow file directory	Enter the location of overflow files written by target tables in this datastore. A variable can also be used.

Session

Option	Description
Additional session parameters	Additional session parameters specified as a valid SQL statement. Delimit multiple SQL statements with semicolons.

Related Information

[Enable Automatic Data Transfer \[page 70\]](#)

5.3.7.3.20 SAP ASE

Create an SAP ASE database datastore to use in Data Services.

Main window

Option	Description
CDC options	Select an option from the dropdown list. Options include the following: <ul style="list-style-type: none">• No CDC: Disables change data capture.• Replication Server CDC: Uses the Replication Server change data capture.
Database version	Select the version of your SAP ASE client. This is the version of SAP Sybase that this datastore accesses.

Option	Description
Database server name	Enter the name of the computer where the SAP ASE instance is located. <div><div>i Note</div><div>For Unix Job Servers, when you logon to a SAP Sybase repository in the Designer, the case in which you type the database server name must match the associated case in the SYBASE_Home\interfaces file. If the case does not match, you might receive an error because the Job Server cannot communicate with the repository.</div></div>
Database name	Enter the name of the database to which the datastore connects.
User name	Enter the user name of the account through which the software accesses the database.
Password	Enter the password associated with the User name .
General	
Option	Description
Rows per commit	Enter the maximum number of rows loaded to a target table before saving the data. The default is 1000. This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.
Overflow file directory	Enter the location of overflow files written by target tables in this datastore. You can enter a variable for this option.
Locale	
Option	Description
Language	For more information, see Locales and Multi-byte Functionality [page 1378] .
Code page	
Server code page	
Session	
Option	Description
Additional session parameters	Additional session parameters specified as a valid SQL statement. Delimit multiple SQL statements with a semicolon.
Aliases (Click here to create)	
Option	Description
Aliases	Click to open the Create New Alias popup dialog. Enter an alias name, click <i>OK</i> , and enter the owner name in the Alias text row.

Related Information

[Enable Automatic Data Transfer \[page 70\]](#)

[Change data capture \(CDC\) \[page 70\]](#)

5.3.7.3.21 Sybase IQ

Create a Sybase IQ (now known as SAP IQ) database datastore to use in Data Services.

Main window

Option	Description
Database version	Select the version of SAP IQ that this datastore accesses. Displayed options in the rest of the datastore editor vary depending on the version selected.
Use data source name (DSN)	<p>Select to use DSN to connect to the database. When you select to use DSN, complete the Data source name option.</p> <p>By default, this option is not selected and the software uses a server name (also known as DSN-less) connection. For a DSN-less connection, complete the following options:</p> <ul style="list-style-type: none">• Database server name• Database name• Port
Hostname	<p>Enter the computer name or IP address.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Database name	<p>Enter the name of the database defined in SAP IQ.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Port	<p>Enter the number of the database port.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Server name	<p>Enter the SAP IQ sever name.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Data source name	<p>Select or type the DSN defined in the ODBC Administrator for connecting to your database.</p> <p>This option is required when you select Use data source name (DSN).</p>
User name	Enter the user name of the account through which the software accesses the database.
Password	Enter the password associated with User name .

General

Option	Description
Rows per commit	<p>Enter the maximum number of rows loaded to a target table before saving the data. The default is 1000.</p> <p>This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.</p>

Option	Description
Bulk loader directory	Enter the location where command and data files are written for bulk loading.
<div> i Note For Solaris systems, the path name must be fewer than 80 characters. </div> <p>You can enter a variable for this option.</p> <p>If you do not enter a name here, Data Services writes the files to the default directory <code><DS_COMMON_DIR>\log\bulkloader</code>.</p>	

Overflow file directory	Enter the location of overflow files written by target tables in this datastore. You can enter a variable for this option.
---	--

Locale

Option	Description
Language	For more information, see Locales and Multi-byte Functionality [page 1378] .
Code page	
Server code page	

Bulk loader

Option	Description
JS and DB on same machine	<p>For some versions of SAP IQ, configure Data Services to transfer the data file generated on the Job Server (JS) to the database server (DB) via FTP.</p> <p>Refer to the following Supported bulk loader options and methods table. It contains information about how to use this option together with Use named pipe and FTP, depending on the version of SAP IQ you use.</p>
Server working directory	<p>Type the path to the working directory for the load utility on the computer that runs the SAP IQ server. If an absolute file path is specified for the FTP host working directory box, then the Server working directory is optional. If the box is left blank, then the software uses the file path to the FTP host working directory.</p> <p>If a virtual file path is specified for the FTP host working directory, enter an absolute file path in the Server working directory box.</p>
Use named pipe	<p>Select Yes to eliminate the need to write a data file to disk, which can improve performance. If a data file is required for SAP IQ database recovery, select No. Defaults to No.</p> <p>Refer to the following Supported bulk loader options and methods table. It contains information about how to use this option together with JS and DB on same machine and FTP, depending on the version of SAP IQ you use.</p>

Supported bulk loader options and methods

JS and DB on same machine	Use named pipe	For SAP IQ versions earlier than 15.x	For SAP IQ versions 15.x and later
Yes	Yes	Named pipe	Named pipe

JS and DB on same machine	Use named pipe	For SAP IQ versions earlier than 15.x	For SAP IQ versions 15.x and later
Yes	No	File	File
No	Yes	FTP	Named pipe
No	No	FTP	File

FTP

Option	Description
FTP host name	For some versions of SAP IQ, Data Services generates a data file and transfers it via FTP to the database for loading. Type the name of the SAP IQ server computer (host). If you leave the FTP host name blank and Data Services needs this FTP information for bulk loading, it generates a validation error.
FTP login user name	Must be defined to use FTP.
FTP login password	Must be defined to use FTP.
FTP host working directory	The location on the database server where Data Services transfers the data file between the Job Server and the SAP IQ server. For Windows servers only, configure a path to a virtual directory.

Session

Option	Description
Additional session parameters	Additional session parameters specified as a valid SQL statement. Delimit multiple SQL statements with semicolons.

SAP Sybase Miscellaneous

Option	Description
Use linked remote servers	Uses a remote server when you use the <code>INSERT_LOCATION</code> SQL statement for a data flow that uses SAP IQ as the loader and SAP ASE or SAP IQ as the reader. Data Services pushes down the SQL statement for the SAP IQ server location. <ul style="list-style-type: none"> Yes: Use remote servers that have already been linked. No: Do not use remote servers that have already been linked. <p>To define a remote server, use the <code>CREATE SERVER</code> SQL statement in SAP IQ. To set up the remote logon for users, use the <code>CREATE EXTERNLOGIN</code> SQL statement.</p> <p>For detailed information about the SQL statements, see the relevant SAP IQ product documentation.</p>

Aliases (Click here to create)

Option	Description
Aliases	Click to open the Create New Alias popup dialog. Enter an alias name, click OK , and enter the owner name in the Alias text row.

Related Information

[Enable Automatic Data Transfer \[page 70\]](#)

5.3.7.3.22 SAP Vora datastore

Access table data in SAP Vora using an SAP Vora datastore as a source or target in a data flow.

The following tables describe the options in the Datastore editor specific to SAP Vora.

SAP Vora datastore options

Option	Description
Database Version	Select your SAP Vora version. i Note You can use only Vora version 2.0 and later for Vora datastores.
Data Source Name	Required for Vora. If you have already created a DSN, select it from the drop-down list. To create a new DSN, select ODBC Admin to open the ODBC Administrator, where you can create a DSN. i Note The driver must be SAP HANA version 2.0 SP2 ODBC or later version
ODBC Admin	Click to open the ODBC Administrator to create a DSN.
User Name	Enter the user name to access SAP Vora.
Password	Enter the password associated with User Name .
Connection	
Option	Description
Additional Connection Parameters	Enter information for any additional connection parameters. Use the format <code><parameter1>=<value1>;</code> <code><parameter2>=<value2></code> .

General

Option	Description
<i>Rows per commit</i>	<p>Enter the maximum number of rows loaded to a target table before saving the data. The default is 1000.</p> <p>This value is the default commit size for target tables in this datastore. You can overwrite this value for individual target tables.</p>
<i>Bulk loader directory</i>	<p>Enter or browse to the location where the software writes command and data files for bulk loading. You can use a variable for this option.</p> <div>i Note<p>You can override this location when you configure the target file for loading.</p></div>

Session

Option	Description
<i>Additional session parameters</i>	<p>Additional session parameters specified as a valid SQL statement. Delimit multiple SQL statements with semicolons.</p>

5.3.7.3.22.1 About SAP Vora datastore

With an SAP Vora datastore, access Vora tables by using the SAP HANA ODBC driver and the SAP HANA wire protocol.

Use the SAP Vora datastore as a source in a data flow, and a template table for the target. For data loading, the software loads data from the template table to a CSV staging file in a local configured file, local HDFS, or a local Amazon S3 HDFS. The software then loads the table from the local file and appends data to the existing table in SAP Vora.

Perform the following tasks with the SAP Vora datastore:

- Import Vora tables.
- Append data to existing Vora tables using INSERT.
- Utilize bulk loading.
- View Vora table data in SAP Data Services.
- Browse metadata.

Consider the following limitations when you use a SAP Vora datastore:

- The datastore does not work for SAP Vora views and partitions.
- The datastore uses the SAP Vora relational disk engine. It is not applicable for other engines such as SAP Vora graph engine or collection engine.

- The datastore does not permit partial column mapping.


SAP Vora datastore requirements:

- Use with SAP Vora version 2.0 and later versions. To access SAP Vora with versions earlier than 2.0, use the ODBC datastore.
- Use the SAP HANA version 2.0 Support Package 2 ODBC driver for the SAP HANA wire protocol.
- The datastore user must be registered as a SAP Vora “Vora user.” For details about user types, see the *SAP Vora Developer Guide*.

5.3.7.3.22.2 Configuring DSN for SAP Vora on Unix and Linux

SAP Vora requires a DSN type connection.

Download and install the SAP HANA ODBC driver version 2.0 SP2 and later. The file name is `libodbcHDB.so`.

To use the graphical user interface for Connection Manager, you must install the GTK+2 library. The GTK+2 is a free multi-platform toolkit that creates user interfaces. For more information about obtaining and installing GTK+2, see <https://www.gtk.org/> . The following instructions assume that you have the user interface for Connection Manager.

1. In a Command Prompt, open the Connection Manager as follows:

Sample Code

```
$ cd $LINK_DIR/bin/
$ ./DSConnectionManager
```

The *SAP Data Services Connection Manager* dialog box opens.

2. In the *Data Sources* tab, select SAP Vora and click *Add*.

The *Configuration for SAP Vora* dialog box opens. If you have downloaded and installed the SAP HANA ODBC driver, the software automatically completes the first option, *ODBC INI File*. If you haven't installed the driver, this option is blank. Go back and install the driver and follow these steps again.

3. Enter the remaining options as described in the following table.

Driver options

Option	Description
<i>DSN Name</i>	Select the DSN name from the dropdown arrow.
<i>User Name</i>	Enter the user name to access the SAP Vora table.
<i>Password</i>	Enter the password to access the SAP Vora table.
<i>Driver</i>	Enter the location and name of the SAP Hana ODBC driver. Name: <code>libodbcHDB.so</code> .
<i>Host Name</i>	Enter the server name.

Option	Description
Port	Enter the port number.

- Optional. Click [Test Connection](#). When the connection is successful, click [OK](#).
- Click [Close](#) to close the Connection Manager.

Related Information

[Configuring DSN for SAP Vora on Windows \[page 124\]](#)

[SAP Vora datastore \[page 121\]](#)

5.3.7.3.22.3 Configuring DSN for SAP Vora on Windows

SAP Vora datastores require a DSN type connection.

Download and install the SAP HANA ODBC driver version 2.0 SP2 and later. The file name is `libodbcHDB.exe`.

Open the ODBC Administrator. Open the administrator from the Datastore editor, or open it from your Start menu.

- Open the [System DSN](#) tab and select [vora2](#).
- Click [Configure](#).

The [SAP HDB \(FULL_BUILD_VERSION\)](#) dialog box opens.

- Enter a name in [Data Source Name](#). Optionally enter a description in [Description](#).
- Enter the server name and port number separated with a colon in [Server:Port](#).

❖ Example

[vora:30115](#)

- Click [Connect](#) to test the connection. When you are finished, click [OK](#).

Related Information

[Configuring DSN for SAP Vora on Unix and Linux \[page 123\]](#)

[SAP Vora datastore \[page 121\]](#)

5.3.7.3.23 Teradata

Create a Teradata database datastore to use in Data Services.

Main window

Teradata option	Description
Database version	Select the applicable client version from the dropdown list for the Teradata that you are using.
Use data source name (DSN)	<p>Select to use data source name (DSN) to connect to the database. When you select DSN, also complete the following option:</p> <ul style="list-style-type: none">• Data Source Name <p>By default, this option is not selected and the software uses a server name (also known as DSN-less) connection. When you do not select DSN, complete the following options:</p> <ul style="list-style-type: none">• Database Server Name• Database Name• Port
Database server name	<p>Enter the name of the computer where the Teradata server is located.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Database name	<p>Enter the database name.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Port	<p>Enter the port number to connect to the Teradata server.</p> <p>This option is required if you did not select Use data source name (DSN).</p>
Data source name	<p>Select an existing DSN from the dropdown list.</p> <p>This option is required when you select Use data source name (DSN).</p>
User name	Enter the applicable user name to access the database.
Password	Enter the password associated with the User Name .

General

Teradata option	Description
Bulk loader directory	Enter or browse to the location where command and data files are written for bulk loading. <div> Note For Solaris systems, the path name must be less than 80 characters. You can enter a variable for this option. </div>
Bulk reader directory	Enter or browse to the location of the directory on the Job Server where you store Teradata Parallel Transporter files (control and error files). If you leave this option empty, the software uses the default location <code><LINK_DIR>\log\BulkReader.</code>
Overflow file directory	Enter or browse to the location of overflow files written by target tables in this datastore. You can use a variable for this option.

Locale

Teradata option	Description
Language	For information, see Locales and Multi-byte Functionality [page 1378] .
Code page	
Server code page	
Teradata	
Log directory	Enter or browse to the directory in which to write log files.
Tdpld	Enter the Teradata Director Program Identifier (Tdpld) that identifies the name of the Teradata database to load. Required when you use bulk loading.

Session

Teradata option	Description
Additional session parameters	Additional session parameters specified as a valid SQL statement. Delimit multiple SQL statements with a semicolon. <div> ❖ Example For example, to use the Table_Comparison transform with Teradata 13 and later tables as the comparison table and target table, make the following settings: <ul style="list-style-type: none"> On the Teradata server, set the General parameter <i>DBSControl</i> to TRUE to allow uncommitted data to be read. In the Data Services Teradata datastore, add the following statement in Additional session parameters: <pre>SET SESSION CHARACTERISTICS AS TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;</pre> </div>

Aliases (Click here to create)

Teradata option	Description
Aliases	Click to open the Create New Alias popup dialog. Enter an alias name, click <i>OK</i> , and enter the owner name in the Alias text row.

Related Information

[Enable Automatic Data Transfer \[page 70\]](#)

5.3.7.4 Adapter datastores

Adapter datastores provide access to the application data that is accessed through the adapter.

Depending on the adapter implementation, adapters allow you to:

- Browse application metadata
- Import application metadata into a repository
- Move batch and real-time data between the software and applications

SAP offers an Adapter Software Development Kit (SDK) to develop your own custom adapters. Also, you can buy the software prepackaged adapters to access application metadata and data in any application. For more information on these products, contact your SAP sales representative.

Adapters are represented in Designer by adapter datastores. Jobs provide batch and real-time data movement between the software and applications through an adapter datastore subordinate objects:

Subordinate Objects	Use as	Used for
Tables	Source or target	Batch data movement
Documents	Source or target	
Functions	Function call in query	
Message functions	Function call in query	Real-time data movement
Outbound messages	Target only	

Adapters can provide access to an application data and metadata or just metadata. For example, if the data source is SQL-compatible, the adapter might be designed to access metadata. The software extracts data from or loads data directly to the application.

For detailed information about installing, configuring, and using adapters, see *Supplement for Adapters*.

Parent topic: [Datastore \[page 56\]](#)

Related Information

[Datastore editor \[page 57\]](#)

[Application datastores \[page 66\]](#)

[Database datastores \[page 68\]](#)

[Web service datastores \[page 128\]](#)

5.3.7.5 Web service datastores

Web service datastores represent a connection from Data Services to an external web service-based data source.

For more information, see “Accessing a web service using the Designer” in the *Integrator Guide*.

Parent topic: [Datastore \[page 56\]](#)

Related Information

[Datastore editor \[page 57\]](#)


[Application datastores \[page 66\]](#)

[Database datastores \[page 68\]](#)

[Adapter datastores \[page 127\]](#)


5.3.8 Document

Documents are available for some adapter datastores to describe a data schema.

Characteristic	Description
	Document icon.
Class	Reusable
Access	In the object library, click the Datastores tab.
Description	<p>Available in some adapter datastores, documents describe a data schema. Documents can support complicated nested schemas. You can use documents as sources or targets.</p> <p>See your adapter's documentation for specific information about the options available for documents.</p>

5.3.9 DTD

A DTD (document type definition) contains a description of a data schema for an XML message or file.

Characteristic	Description
	DTD icon.
Class	Reusable
Access	In the object library, click the <i>Formats</i> tab, then open the Nested Schemas category.
Description	<p>A DTD (document type definition) describes the data schema of an XML message or file.</p> <div><p>i Note</p><p>XML Schemas can be used for the same purpose.</p></div> <p>Data flows can read and write data to messages or files based on a specified DTD format. You can use the same DTD to describe multiple XML sources or targets.</p> <p>To use DTDs, import metadata into SAP Data Services. You can import a DTD directly, or you can import an XML document that contains or references a DTD. During import, the software converts the structure defined in the DTD into the nested-relational data model (NRDM).</p>

[Editor \[page 130\]](#)

Edit a DTD to enhance performance or to tune the nested relational data models.

[DTD Properties \[page 130\]](#)

DTD properties include values that you cannot change, and values that you can enable, disable, or edit.

[Attributes for DTDs \[page 132\]](#)

DTDs have two groups of attributes: Column and nested table.

[Supported DTD components \[page 133\]](#)

The software supports the declarations in a DTD component based on the content model.

[Rules for importing DTDs \[page 136\]](#)

SAP Data Services uses conversion rules to convert a DTD to an internal schema.

[Design considerations \[page 137\]](#)

Use design techniques to improve performance and tune nested relational data models for DTDs.

[Error checking \[page 139\]](#)

Control whether SAP Data Services checks an incoming XML file or message for validity.

Related Information

[XML schema \[page 286\]](#)

5.3.9.1 Editor

Edit a DTD to enhance performance or to tune the nested relational data models.

Open the DTD editor by double-clicking a DTD name in the object library.

Parent topic: [DTD \[page 129\]](#)

Related Information

[DTD Properties \[page 130\]](#)

[Attributes for DTDs \[page 132\]](#)

[Supported DTD components \[page 133\]](#)

[Rules for importing DTDs \[page 136\]](#)

[Design considerations \[page 137\]](#)

[Error checking \[page 139\]](#)

5.3.9.2 DTD Properties

DTD properties include values that you cannot change, and values that you can enable, disable, or edit.

Context-click a DTD file under Nested Schemas in the [Formats](#) tab of the object library and select [Properties](#). There are two tabs in the DTD properties dialog: [General](#) and [Format](#).

General tab

Property	Description
Name	(Read only) The name of the DTD file format. This name appears in the object library under the Nested Schemas category of the Formats tab and is used for sources and targets (XML files or messages) that reference this format in data flows.
Description	Text that describes the DTD. This text is entered when you create the DTD format file.

Format tab

Property	Description
<i>File location</i>	(Optional) Select the name of an existing file location object. The file location object contains file transfer protocol information and local and remote server information to safely transfer data from remote to local server (source) and local to remote server (target).
<i>Delete file after transfer</i>	Available when you select a file location object above. As a source: <ul style="list-style-type: none">• Not applicable for SCP file transfer protocol. Check to delete the local file copy after the software loads it as a source in the data flow.• Uncheck to save the local file copy after the software loads it as a source in the data flow. As a target: <ul style="list-style-type: none">• Not applicable for SCP file transfer protocol. Check to delete the local file copy after the software transfers the generated output data from the local to the remote file.• Uncheck to save the local file copy after the software transfers the generated output data from the local to the remote file.
<i>File name</i>	Select the DTD file name from the drop list or click <code><select file></code> to browse for the DTD file. For added flexibility, select a variable for this option.
<i>Imported from</i>	(Read only) Contains the full path to the DTD file.
<i>DTD file</i>	(Read only) If the check box is selected, the DTD format was originally imported from a DTD file. Otherwise, it was imported from an XML file with an associated DTD.
<i>Root element name</i>	(Read only) The name of the primary node of the XML that the DTD is defining. SAP Data Services only imports elements of the format that belong to this node or any sub nodes.

Parent topic: [DTD \[page 129\]](#)

Related Information

[Editor \[page 130\]](#)

[Attributes for DTDs \[page 132\]](#)

[Supported DTD components \[page 133\]](#)

[Rules for importing DTDs \[page 136\]](#)

[Design considerations \[page 137\]](#)

[Error checking \[page 139\]](#)

5.3.9.3 Attributes for DTDs

DTDs have two groups of attributes: Column and nested table.

View the DTD attributes tab in the Properties dialog. The following DTD attributes are supported.

Supported column attribute	Description
Enumeration	Contains a list of all possible values separated by vertical bars. For example: "Red White Blue Green Magenta". A string display is cut off at 256 characters.
Fixed Value	The only value the column can have.
Native Type	String. The original data type of the element or attribute in the DTD.
Required	Indicates whether this column always has to be mapped (YES/NO). If a column is optional (required =no), then validation will allow mapping expressions to be missing for these columns and at runtime the engine will substitute NULLs for the missing values.
XML Type	Allows you to track whether the column was an element or attributes in the original DTD.

Supported nested table attribute	Description
Any One Column	If choice (for example, "white black almond"), then SAP Data Services sets the value of Any One Column to YES. If sequence (for example, "first, last, street, city, state") then the software sets the value to NO. If both are present in the DTD, the value is set to NO.
Minimum Occurrence	If ()* then minimum occurrence is set to zero. If ()+, then minimum occurrence is set to 1. Indicates minimum number of rows that can be in the table.

Parent topic: [DTD \[page 129\]](#)

Related Information

[Editor \[page 130\]](#)

[DTD Properties \[page 130\]](#)

[Supported DTD components \[page 133\]](#)

[Rules for importing DTDs \[page 136\]](#)

[Design considerations \[page 137\]](#)

[Error checking \[page 139\]](#)

5.3.9.4 Supported DTD components

The software supports the declarations in a DTD component based on the content model.

SAP Data Services reads the following DTD components. To process the data read in an XML file or message, the software translates the DTD into its internal nested-relational database model.

Each component in the DTD is defined by its `<content model>`. The software supports the declarations in XML content models as follows:

DTD declaration		Supported
DOCTYPE	SYSTEM	Supported.
	PUBLIC	No support.
Declarations	ELEMENT	Supported. The XML Type attribute of the corresponding column is set to Element.
	ATTRIBUTE	Supported. The XML Type attribute of the corresponding column is set to Attribute.
	ENTITY	Supported. All entity references that can be expanded are expanded. Any that cannot be expanded cause an error at the time that you import the DTD.
	NOTATION	No support. Elements defined with NOTATION cause an error at the time that you import the DTD.
Content model	ANY	No support. Elements defined with ANY cause an error at the time that you import the DTD.
	EMPTY	Supported.
	#PCDATA	Supported. Converts to varchar(1024).
	MIXED	Supported.
Attribute declarations	CDATA	Supported. Converts to varchar(1024).
	ID	Supported. Converts to varchar(1024). When producing XML output, the software cannot ensure that ID values are unique throughout the schema.
	IDREF	Supported. Converts to varchar(1024).
	IDREFS	Supported. Converts to varchar(1024).

DTD declaration	Supported
NMTOKEN	Supported. Converts to <code>varchar(1024)</code> .
NMTOKENS	Supported. The software treats multiple tokens as a single token with more than one space-separated values.
Enumerated value	<p>Supported. The software saves the enumerated values in the Enumeration attribute of the column.</p> <p>When producing XML output, the software checks to ensure that the value generated by the real-time job for the corresponding column is from the list; if no value is generated, the software uses the provided default value.</p> <p>If you validate XML messages against the DTD in a real-time job and the message includes a value that is not allowed based on the DTD, the XML source produces an error.</p>
Attribute declaration defaults	<p>#REQUIRED</p> <p>Supported.</p> <p>The software saves this as the Required attribute with a value of YES and as data type <code>varchar (1024)</code> . When producing XML output, the software always provides a value. If there is no value supplied, the output value is NULL (' ').</p>
	<p>#IMPLIED</p> <p>Supported.</p> <p>The software saves this as the Required attribute with a value of NO and as the data type <code>varchar (1024)</code> . When producing XML output, the software provides whatever value is generated in the data flow for the corresponding column, including a NULL value (' ').</p>
	<p>#FIXED (default value)</p> <p>Supported.</p> <p>The software saves this as the Fixed Value attribute and the data type <code>varchar (1024)</code> . When producing XML output, the software checks to ensure that the value generated by the real-time job for the corresponding column is from the list; if no value is generated, the software uses the provided default value.</p>
	<p>Default values</p> <p>Supported.</p> <p>Converts to data type <code>varchar (1024)</code> . When producing XML output, the software uses the default value if the value generated in the real-time job for the corresponding column is NULL.</p>

To produce a data model that can include all possible configurations of an element, the software can simplify some of the content model operations:

Operator	Description	Supported
No operator	One and only one	One and only one.

Operator	Description	Supported
Comma (,)	Sequence	Supported. The software uses the ordering given in the DTD as the column ordering in the internal data set. Also the Any One Column attribute is set to a value of NO.
Vertical bar ()	Choice (either/or)	Supported. The software uses the ordering given in the DTD as the column ordering in the internal data set. Also the Any One Column attribute is set to a value of YES. The internal data set must include both options.
Plus (+)	One or more	Supported. Saved as nested table attribute Minimum Occurrence with a value of "1". The internal data set must include options for one or more elements.
Asterisk (*)	Zero or more	Supported. Saved as nested table attribute Minimum Occurrence with a value of "0". The software translates an item or grouping including zero or more items into a nested table.
Question mark (?)	Optional	Supported. The internal data set includes the Required attribute set to a value of NO for the corresponding column or nested table.
Parentheses ()	Group	Dropped. The internal data set does not maintain groupings unless the group is operated on by the * operator. If the group can allow more than one item, the software makes a new nested table into which it places the elements in the group.

After these simplifications, the software needs only work with two DTD operators: sequence (strict ordering) and the combined operators of the group operator with the zero or more item operator. For the purpose of representing the data internally in the software, all DTDs can now be written using only , or ()*.

Parent topic: [DTD \[page 129\]](#)

Related Information

[Editor \[page 130\]](#)

[DTD Properties \[page 130\]](#)

[Attributes for DTDs \[page 132\]](#)

[Rules for importing DTDs \[page 136\]](#)

[Design considerations \[page 137\]](#)

[Error checking \[page 139\]](#)

5.3.9.5 Rules for importing DTDs

SAP Data Services uses conversion rules to convert a DTD to an internal schema.

SAP Data Services applies the following rules to convert a DTD to an internal schema:

- Any element that contains an PCDATA only and no attributes becomes a column.
- Any element with attributes or other elements (or in mixed format) becomes a table.
- An attribute becomes a column in the table corresponding to the element it supports.
- Any occurrence of choice (.) or sequence (|) operators uses the ordering given in the DTD as the column ordering in the internal data set.
- Any occurrence of a multiple entities, such as (*) or (+), becomes a table with an internally generated name (an implicit table).
- The internally generated name is the name of the parent followed by an underscore, then the string "nt" followed by a sequence number. The sequence number starts at 1 and increments by 1.

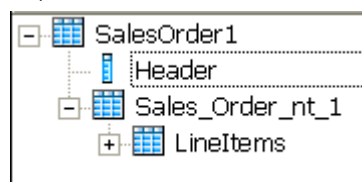
After applying these rules, the software uses two additional rules, except where doing so would allow more than one row for a root element:

- If an implicit table contains one and only one nested table, then the implicit table can be eliminated and the nested table can be attached directly to the parent of the implicit table.

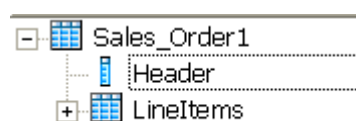
For example, the SalesOrder element might be defined as follows in a DTD:

```
<!ELEMENT SalesOrder (Header, LineItems*)>
```

When converted into the software, the LineItems element with the zero or more operator would become an implicit table under the SalesOrder table. The LineItems element itself would be a nested table under the implicit table.



Because the implicit table contains one and only one nested table, the format would remove the implicit table.

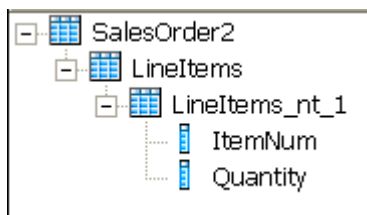


- If a nested table contains one and only one implicit table, then the implicit table can be eliminated and its columns placed directly under the nested table.

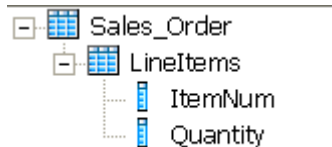
For example, the nested table LineItems might be defined as follows in a DTD:

```
<!ELEMENT LineItems (ItemNum, Quantity)*>
```

When converted into the software, the grouping with the zero or more operator would become an implicit table under the LineItems table. The ItemNum and Quantity elements would become columns under the implicit table.



Because the LineItems nested table contained one and only one implicit table, the format would remove the implicit table.



Parent topic: [DTD \[page 129\]](#)

Related Information

[Editor \[page 130\]](#)

[DTD Properties \[page 130\]](#)

[Attributes for DTDs \[page 132\]](#)

[Supported DTD components \[page 133\]](#)

[Design considerations \[page 137\]](#)

[Error checking \[page 139\]](#)

5.3.9.6 Design considerations

Use design techniques to improve performance and tune nested relational data models for DTDs.

The following areas provide opportunities for you to improve performance and tune the nested-relational data model results for a given DTD:

- **Recursion**
If the DTD contains an element that uses an ancestor element in its definition, SAP Data Services expands the definition of the ancestor for a fixed number of levels. For example, given the following definition of element "A":
A: B, C
B: E, F
F: A, H
The software produces a table for the element "F" that includes an expansion of "A." In this second expansion of "A," "F" appears again, and so on until the fixed number of levels. In the final expansion of "A," the element "F" appears with only the element "H" in its definition.
- **Repeated column names**
The software does not allow more than one column with the same name at the same level in a table. If the internal schema that the software produces from a DTD contains duplicate names, the software adds a

suffix to each instance of the duplicated name to ensure unique column names. The suffix is an underscore followed by a sequence number, starting at 1 and incrementing by 1.

A DTD can produce duplicate names when the DTD contains a repeated element at one level or an element with a scalar value with an attribute of the same name.

- **Ambiguous DTDs**
You can create a DTD such that the software does not have enough information to make a unique decision when generating the internal data set. The software reacts to an ambiguous DTD by throwing an error for the XML message source at run time. An example of an ambiguous definition is as follows:

DTD	A: ((B, (C) *) (B, (D) *)) +
Schema in Data Services	A: (B, (C) *, B, (D) *) *
XML input	A: text <D>1</D> <D>2</D>

The software will use the B element data to populate the first B column, then use the D element data to populate the D element. If this data is then translated back into XML, it would be invalid relative to the DTD.

Metadata

If you delete a DTD from the object library, XML file and message sources or targets that are based on this format are invalid. The software marks the source or target objects with an icon that indicates the calls are no longer valid.

Sample_order



To restore the invalid objects, you must delete the source or target and replace it with a source or target based on an existing DTD.

Parent topic: [DTD \[page 129\]](#)

Related Information

- [Editor \[page 130\]](#)
- [DTD Properties \[page 130\]](#)
- [Attributes for DTDs \[page 132\]](#)
- [Supported DTD components \[page 133\]](#)
- [Rules for importing DTDs \[page 136\]](#)
- [Error checking \[page 139\]](#)

5.3.9.7 Error checking

Control whether SAP Data Services checks an incoming XML file or message for validity.

If you choose to check each XML file or message, the software uses the DTD imported and stored in the repository rather than a DTD specified by a given XML file or message. If a file or message is invalid relative to the DTD, the real-time job produces an error and shuts down.

During development, you might validate all files and messages to test for error conditions. During production, you might choose to accept rare invalid files or messages and risk ambiguous or incorrect data.

All files or messages that the software produces for an XML file or message target are validated against the imported DTD.

You can enable or disable validation for an XML file or message source or target in that object's editor.

Parent topic: [DTD \[page 129\]](#)

Related Information

[Editor \[page 130\]](#)

[DTD Properties \[page 130\]](#)


[Attributes for DTDs \[page 132\]](#)

[Supported DTD components \[page 133\]](#)

[Rules for importing DTDs \[page 136\]](#)

[Design considerations \[page 137\]](#)

5.3.10 Excel workbook format

Characteristic	Description
	Excel workbook format icon.
Class	Reusable
Access	In the object library, click the <i>Formats</i> tab.
Description	An Excel workbook format describes the structure defined in an Excel workbook (denoted with a .xls extension). You store format templates for Excel data ranges in the object library. You use the template to define the format of a particular source in a data flow. SAP Data Services accesses Excel workbooks as sources only (not as targets).

5.3.10.1 Defining format schema

You can define the format schema by:

- You can manually define (enter) column names, data types, content types and descriptions in the schema pane at the top of the window.
- Import the Excel metadata from a sample Excel workbook and create the schema automatically. You can opt to import the schema from:
 - A named range defined in an Excel workbook.
 - A custom range in a worksheet (for example A1:C10).
 - All fields. Note that the field range does not necessarily begin on row 1, column 1; it starts at the upper left-most cell in the worksheet that contains data.

After you select an access method, import the schema by clicking Import schema.

i Note

Importing overwrites the existing schema.

After you import, you can edit column names, data types, and content types in the schema pane at the top of the window. You can optionally add descriptions for each column.

You will see Blank if the content type cannot be automatically determined for a column.

5.3.10.2 Notes

- If SAP Data Services cannot determine the data type for a column, for example if your column selection doesn't contain any data, it imports the column as varchar(255).
- If the worksheet is empty and you select the *All fields* option, the software creates a single field, F1 varchar(255).
- For workbook-specific (global) named ranges, the software would name a range called *range* as *range*. However for worksheet-specific (local) named ranges, the software would name a range called *range* that belongs to the worksheet *Sheet1* as *range!Sheet1*.
In UNIX, you must also include the worksheet name when defining a workbook-specific (global) named range.
- Because the software reads stored formula values, incorrect values could result if Excel does not exit properly. Close and reopen the file in Excel to refresh the values.
- If an invalid Excel formula displays an error such as #DIV/0, #VALUE or #REF!, the software processes the cell as NULL.
- You cannot import or process a password-protected workbook.
- The software might not be able to import or process blank worksheet names or those that contain special characters such as /?:!*[]`.

5.3.10.3 Import or Edit Excel workbook format options

Descriptions for the Import or Edit Excel workbook dialog and options.

The top panel of the [Import](#) or [Edit Excel Workbook](#) window displays the following information:

Display	Description
Format name	The name of the Excel workbook format template in the object library. In the Import Excel Workbook window, you can define the name of the format.
Schema pane	Manually define (or edit) the schema of the Excel workbook template.

The Import (and Edit) Excel workbook format dialog boxes include options on the [Format](#) and [Data Access](#) tabs.

Related Information

[Format tab \[page 141\]](#)

[Data Access tab \[page 144\]](#)

5.3.10.3.1 Format tab

Option descriptions for the Import or Edit Excel workbook [Format](#) tab.

The [Format](#) tab defines the parameters of the Excel workbook format.

Format option	Description
File location	<p>(Optional) Select the name of an existing file location object.</p> <p>The file location object contains file transfer protocol information (such as FTP or SFTP) and local and remote server information. The software uses this information to transfer a copy of the Excel workbook file from a remote server to your local server to use as a source in a data flow.</p> <div><p>i Note</p><p>When this option is enabled, the Directory option is disabled, and the Data Access tab is also disabled.</p></div>

Format option	Description
<i>Delete file after transfer</i>	<p>Available when you select a file location object.</p> <ul style="list-style-type: none"> • Check to delete the local file copy after the software loads it as a source in the data flow. • Uncheck to save the local file copy after the software loads it as a source in the data flow.
<i>Directory</i>	<p>Specify the directory that contains the Excel workbook. You can specify variables or wild cards, but the <i>Import schema</i> button will then be disabled.</p> <div> <p>i Note</p> <p>If you enter a file location object name in <i>File Location</i>, this option is disabled.</p> </div>
<i>File name</i>	<p>Specify the file name for the Excel workbook file. This file contains the schema definition. You can specify variables or wild cards, but the <i>Import schema</i> button will then be disabled.</p>

Format option	Description
<i>Access method</i>	<p>Specifies whether to import a named range (as defined in Excel) or a range from a worksheet:</p> <p><i>Named range</i>—The drop-down box displays named ranges defined in the Excel workbook (provided that the directory/file name combination refers to a valid file). You can also type the name manually.</p> <div> <p>i Note</p> <p>The drop-down does not display non-contiguous named ranges, even if they exist in the Excel workbook.</p> </div> <p><i>Worksheet</i>—You can select a specific worksheet by either name or ordinal number (select the <i>Number</i> check box to designate the worksheet name as a number). The drop-down box displays all worksheets in the Excel workbook (provided that the directory/filename combination refers to a valid file). You can also type the name manually.</p> <div> <p>i Note</p> <p>If the worksheet name starts with a dollar sign (\$), Data Services treats it as a variable; to use a worksheet name that starts with a dollar sign, prefix (escape) it with a backslash (\).</p> </div> <p>The Worksheet method includes the following <i>Range</i> options:</p> <ul style="list-style-type: none"> • <i>All fields</i>—Includes everything from the uppermost left-hand populated cell to the lowest, right-hand cell. • <i>Custom range</i>—Uses Excel notation, for example A1:B3. Click the button to the right of the field to launch a new instance of Excel (if installed) to select the cell range directly in the worksheet. <div> <p>i Note</p> <p>Subsequently making the software the active window closes this Excel instance.</p> </div> <p><i>Extend range</i>—If you are using a custom range, select this check box to extend the custom range selection to the end of the worksheet.</p>
<i>Code page</i>	<p>Specifies the character encoding (code page) of the character data in the Excel workbook. For more information, see the Locales and Multi-Byte Functionality section in the <i>Reference Guide</i>.</p>
<i>Use first row values as column names</i>	<p>Select to use the first-row values as column names. If this check box is cleared, then the software names the fields for you (F1 represents the first field, F2 represents the second field, and so on).</p>

5.3.10.3.2 Data Access tab

Option descriptions for the Import or Edit Excel workbook [Data Access](#) tab.

The Data Access tab specifies how the software accesses the data file. If both the [FTP](#) and [Custom](#) check boxes are cleared, the software assumes the data file is on the same computer as the Job Server.

→ Tip

Consider using a file location object instead of defining data access in the Data Access tab. A file location object allows greater variety of file transfer methods, and provides for editing outside of the data flow.

Data access option	Description
FTP	Select to use FTP to access the data file.
Host	Type the computer (host) name, fully qualified domain name, or IP address of the computer where the data file resides.
User	Type the FTP user name.
Password	Type the FTP user password.
Directory	Type or browse to the directory that contains the Excel workbook data file to import. If you include a directory path here, then enter only the file name in the Name field.
File name	Type or browse to the Excel workbook data file Name . You can use variables or wild cards (* or ?). If you leave Directory blank, then type a full path and file name here.
Custom	Select to use a custom executable to access the data file.
Executable	Type the name of the program to read data file.
User	Type the user name.
Password	Type the password.
Arguments	Include any custom program arguments.

5.3.10.4 Excel workbook source options

Option descriptions for Excel workbook [Source](#) tab.

The Excel workbook source editor includes three tabs: [Source](#), [Format](#), and [Data Access](#). The following tables describe the options in the [Source](#) tab. For option descriptions for the [Format](#) and the [Data Access](#) tabs, see [Format tab \[page 141\]](#) and [Data Access tab \[page 144\]](#).

i Note

Many fields in the source editor allow you to select from a list of variables.

Source option	Description
Make port	Makes the source table an embedded data flow port.
Performance	
Source option	Description
Join rank	<p>Indicates the rank of the source relative to other tables and files joined in a data flow. The software joins sources with higher join ranks before joining sources with lower join ranks.</p> <p>Join rank specified in the Query transform editor FROM tab overrides any join rank specified in a source. For new jobs, specify the join rank only in the Query transform editor.</p> <p>Must be a non-negative integer. Default value is 0.</p> <p>For more information, see the Other Tuning Techniques section in the <i>Performance Optimization Guide</i>.</p>
Cache	<p>Indicates whether the software should read the required data from the source and load it into memory or pageable cache. Because an inner source of a join must be read for each row of an outer source, you might want to cache a source when it is used as an inner source in a join.</p> <p>Options are:</p> <ul style="list-style-type: none">• Yes: The source is always cached unless it is the outer-most source in a join.• No: The source is never cached. <p>The default is Yes.</p> <p>Cache specified in the Query transform editor FROM tab overrides any cache specified in a source. For new jobs, specify the cache only in the Query transform editor.</p>
Skip all empty rows	Select to skip any rows that are empty in the workbook. Clear to also import empty rows, which will display as NULLs.
Error handling	
Source option	Description
Log errors to file	Specifies whether to produce an error report. Defaults to unchecked. Columns with run-time conversion errors will contain NULLs.
Maximum errors to log	If Log data conversion warnings is enabled, you can limit how many warnings the software logs. Defaults to {no limit} .
Error file directory	Type or browse to the directory in which to store the error file.
Error file name	Type or browse to the error file name.

Include file name column


Source option	Description
Include file name column	<p>Determines whether to add a column that contains the source Excel workbook file name in the source output. Defaults to <i>No</i>.</p> <p>Change the value to <i>Yes</i> when you want to identify the source Excel workbook file in situations such as the following:</p> <ul style="list-style-type: none">You specified a wildcard character to read multiple source Excel workbook files at one timeYou load from different source workbook files on different runs
Modify	<p>If the file name is included, this button enables you to modify <i>File name column</i> and <i>Column size</i>.</p>
File name column	<p>If the file name is included, the name of the column that holds the source Excel workbook file name.</p> <p>Defaults to <i>DI_FILENAME</i>.</p>
Column size	<p>If the file name is included, the size (in characters) of the column that holds the source Excel workbook file name.</p> <p>Defaults to <i>100</i>. If the size of the file name column is not large enough to store the file name, truncation occurs from the left.</p>
Include path	<p>If the file name is included, determines whether to include the full path name of the source Excel workbook file.</p> <p>Defaults to <i>No</i>.</p>
Other options	
Source option	Description
Skip first	<p>Optionally enter the number of rows to skip (not read) starting at the top of the worksheet. Defaults to <i>{none}</i>.</p>
Read total	<p>Optionally enter the number of total rows to read starting at the top of the worksheet or after the <i>Skip first</i> value. Defaults to <i>{no limit}</i>.</p>

Related Information

[Format tab \[page 141\]](#)

[Data Access tab \[page 144\]](#)

5.3.11 File format

Characteristic	Description
	File format icon.
Class	Reusable
Access	In the object library, click the Formats tab.
Description	A file format describes the structure of an ASCII file. You store templates for file formats in the object library. You use the templates to define the file format of a particular source or target file in a data flow.

Related Information

[Supported locales and encodings \[page 1397\]](#)

[General file format property options \[page 149\]](#)

[Data File format property options \[page 151\]](#)

[Delimiters format property options \[page 156\]](#)

[Default Format property options \[page 158\]](#)

[Input/Output format property options \[page 160\]](#)

[Custom Transfer format property options \[page 163\]](#)

[Error Handling format property options \[page 164\]](#)

[Source Information format property options \[page 167\]](#)

5.3.11.1 File format properties and modes

Use the [File Format Editor](#) to set properties for file format templates and source and target file formats.

Available properties vary by the mode in which you open the [File Format Editor](#). The following table describes the modes.

Mode	Description
New	Use to create a new file format template. To open in new mode, go to the Formats tab in the object library, right-click Flat Files , and select New . The File Format Editor appears.

Mode	Description
Edit	Use to edit an existing file format template. To open in edit mode, go the Formats tab in the object library, select an existing flat file format, right-click and select Edit . The File Format Editor appears.
Source	Use to edit the file format of a particular source file. To open in source mode, click the name of the source file in the workspace. The File Format Editor appears below the file's schema.
Target	Use to edit the file format of a particular target file. To open in target mode, click the name of the target file in the workspace. The File Format Editor appears below the file's schema.

The file format editor has three work areas:

Work area	Location	Description
Properties-values	The left side of the file format editor window.	Edit the values for file format properties. Expand and collapse the property groups by clicking the leading plus or minus.
Column attributes	The upper right side of the file format editor window.	Edit and define the columns or fields in the file. Field-specific formats override the default format set in the Properties-Values area.
Data preview	The lower right side of the file format editor window.	View how the settings affect sample data, when applicable.

The work area on the left lists groups of file format properties that are not field specific. However, the options vary based on the file format type you choose. The file format property groups include:

- General
- Data File(s)
- Delimiters
- Default Format
- Input/Output
- Locale
- Custom Transfer
- Error Handling
- Source Information

The [Show ATL](#) button displays a view-only copy of the transformation language file generated for your file format.

i Note

You might be directed to use this button by SAP Business User Support.

5.3.11.2 General file format property options

The [General](#) property options are listed in the following table.

The work area in the [File Format Editor](#) lists file format properties that are not field specific.

i Note

See [File format \[page 147\]](#) for mode descriptions.

Option	Value	Description	Mode
Type	Delimited, Fixed width, SAP transport, Unstructured text, Unstructured binary	<p>The format of the data in the text file. Available properties change based on the selected file format type.</p> <p>For information about the SAP transport file format, see the <i>Supplement for SAP</i>.</p>	New, Edit
Name	Any alphanumeric character and underscores	A descriptive name for the file format. This name appears in the object library.	New
Join Rank	Integer greater than or equal to 0	<p>Indicates the rank of the source relative to other tables and files joined in a data flow. The software joins sources with higher join ranks before joining sources with lower join ranks.</p> <p>Join rank specified in the Query transform editor FROM tab overrides any join rank specified in a source. For new jobs, specify the join rank only in the Query transform editor.</p> <p>Must be a non-negative integer. Default value is 0.</p> <p>For more information, see the “Other Tuning Techniques” section in the <i>Performance Optimization Guide</i>.</p>	Source
Cache	Yes, No	<p>Indicates whether the software should read the required data from the source and load it into memory or pageable cache. Because an inner source of a join must be read for each row of an outer source, you might want to cache a source when it is used as an inner source in a join.</p> <p>Options are:</p> <ul style="list-style-type: none">• Yes: The source is always cached unless it is the outer-most source in a join.• No: The source is never cached. <p>The default is Yes.</p> <p>Cache specified in the Query transform editor FROM tab overrides any cache specified in a source. For new jobs, specify the cache only in the Query transform editor.</p>	Source

Option	Value	Description	Mode
Adaptable Schema	Yes, No	<p>Indicates whether the schema of a delimited file format is adaptable or fixed.</p> <ul style="list-style-type: none"> Yes: Indicates that the schema is adaptable. The actual file can contain fewer or more columns than indicated by the file format. If a row contains fewer columns than expected, the software loads null values into the columns missing data. If a row contains more columns than expected, the software ignores the additional data. No: Indicates that the schema is fixed. The software requires the number of columns in each row to match the number of columns specified in the file format. The default is No. If you select Yes, you must ensure that the selected column and row delimiters do not appear inside the actual data. 	
Data Alignment	Character, Byte	<p>Indicates how the software will process fixed-width file formats. The default setting is Character. Character indicates that fields in your data are measured as character. All processing will be in character semantics. Byte indicates that fields in your data are measured as bytes. All processing will be in byte semantics. For example, in character semantic, if you define a column as <code>varchar(30)</code>, it means that the column has 30 characters regardless of the number of bytes for each character. In byte semantic, <code>varchar(30)</code> means 30 bytes. Byte semantic recognizes bytes only, it does not recognize characters.</p> <div> <p>Note</p> <p>The reader reads as many bytes in a column based on the length of the column. For multibyte code pages, a character can be more than one byte. There could be times when trailing bytes of the last character of a column exceeds the length of the column. When this happens the reader continues reading until it gets to the total length of the character.</p> <p>Also in this situation, the loader writes as many bytes as what is specified for the length of the column. The loader then truncates trailing bytes if they exceed the length of the column and the code page is multibyte. Therefore there is a potential for partial characters in the file. When this happens, errors are only logged in the reader. You can specify the error logging options in the File Format Editor.</p> </div>	New, Edit
Make Port	Yes, No	<p>Indicates whether the file is an embedded data flow port. Choose Yes to make a source or target file an embedded data flow port. The default is No.</p> <p>For more information, see “Creating embedded data flows” in the <i>Designer Guide</i>.</p>	

Option	Value	Description	Mode
Rows to read	Integer or blank	Indicates the maximum number of rows that the software reads. The default is blank. If the value is zero or negative, the software reads all rows.	Source
Custom transfer program	Yes, No	<p>Enables the software to use a third-party file transfer program. Displays additional custom transfer program options in the <i>File Format Editor</i> below the Input/Output properties.</p> <div> <p>→ Tip</p> <p>Consider using a file location object instead of a custom transfer program. A file location object allows greater variety of file transfer methods, and provides for editing outside of the data flow. To use a file location object, select No for this option and select the name of an existing file location object in the <i>Location</i> option under <i>Data Files</i>.</p> </div>	All
Skip error handling	Yes, No	Selecting <i>Yes</i> disables the Error Handling section in the Format editor. The default is <i>No</i> .	New, Edit, Source
Parallel process threads	Integer greater than 0, {none}, {default}	<p>Specifies the number of threads for parallel processing, which can improve performance by maximizing CPU usage on the Job Server computer. For example, if you have four CPUs, enter 4 for this option. For more information, see "File multi-threading" in the <i>Performance Optimization Guide</i>.</p> <p>For jobs that process USPS certification tests, the value should be set to <i>{none}</i>.</p>	All

Related Information

[File format \[page 147\]](#)

[Custom Transfer format property options \[page 163\]](#)

[File location object \[page 171\]](#)

5.3.11.3 Data File format property options

Descriptions for options in the *Data file* section of the *File Format Editor*.

The work area on the left, in the *File Format Editor* lists file format properties that are not field specific. The following table contains the *Data file* property options.

Data File format properties

Option	Possible values	Description	Mode
Location	Local, Job Server, or name of an existing file location object	<p>During design: (New or edit modes).</p> <ul style="list-style-type: none"> Select Local to indicate that the data file is located on the local machine that runs the Designer. Select Job Server to indicate that the data file is located on the machine that runs the Job Server. You must enter the absolute path to files. Remember that UNIX systems are case-sensitive. <p>During execution (Source or target modes): All files must be located on the Job Server machine that executes the job. If you use different files to design your job, change the file specified (through the Root directory and File properties options) before execution.</p> <p>(Optional) File location object: (Source or target) To associate a file location object to this file format, select an existing file location object.</p> <p>A file location object contains FTP, SFTP, SCP, and Local protocol information and locations for local and remote servers.</p>	All

Option	Possible values	Description	Mode
<i>Root directory</i>	Path name for the file or blank	<p>The directory where the data file is located.</p> <p>For added flexibility, you can enter a variable for this option.</p> <p>If you enter a directory name, then enter only the file name for the <i>File</i> property.</p> <p>If you leave the root directory blank, then enter a file name that includes the full path name in the <i>File</i> property.</p>	New, Edit

Option	Possible values	Description	Mode
<i>File name(s)</i>	File name(s), file name(s) including full path name, or blank	<p>In new and edit modes, specifies an existing file on which you base this file format description. Data from this file appears in the column attributes area (upper right side) of the File Format editor. In the new and edit modes, you can leave this property blank.</p> <p>In source and target modes, specifies the location of the actual file for this source or target. In source and target modes, you cannot leave this property blank. For added flexibility, you may use the following for this option:</p> <ul style="list-style-type: none"> • A variable that is set to a particular file with full path name. Use variables to specify file names that you cannot otherwise enter, such as file names that contain multi-byte characters. • A list of files, separated by commas, or a file name containing a wild card. In this case, the software reads all these files as a single source. For more information about listing multiple files, see “Reading multiple files at one time” in the <i>Designer Guide</i>. 	All
<i>Read subfolders</i>	Yes, No	For unstructured file formats, specifies whether to read the files in any and all nested subfolders.	New, Edit, Source
<i>Skip empty files</i>	Yes, No	<p>For unstructured file formats, specifies whether to ignore empty files.</p> <ul style="list-style-type: none"> • Yes: Skips empty files. • No: Creates a row with NULL data. 	New, Edit, Source

Option	Possible values	Description	Mode
<i>Number of files to read</i>	Integer or {none}	For unstructured file formats, indicates the maximum number of files to read. A zero or negative value reads all files. The default is {none}.	Source
<i>Delete file</i>	Yes, No	<p>Specifies whether the software should delete the target file before loading generated output data:</p> <ul style="list-style-type: none"> • Yes: Indicates that the software should delete the contents of the target file before loading the newly-generated output data from the data flow. • No: Indicates that the software should append the newly-generated output data to any existing data in the target file. <p>If you have entered a file location object for Location above, the setting affects the target file in the remote server.</p> <ul style="list-style-type: none"> • Yes: Indicates that the software should delete the existing content in the remote target file before loading the newly-generated output data from the local file. • No: Indicates that the software should append the newly-generated output data from the local file to the target file in the remote server. 	Target

Option	Possible values	Description	Mode
Delete file after transfer	Yes, No	<p>Applicable when you choose a file location object for Location above. Specifies to save or delete the data in the local server file.</p> <p>When you use the file format as a source:</p> <ul style="list-style-type: none"> Yes: Deletes local file after software reads data from local file into data flow. No: Saves local file after software reads data from local file into data flow. <p>When you use the file format as a target:</p> <ul style="list-style-type: none"> Yes: Overwrites target local file with generated output data. No: Appends generated output data to the target local file. Applies only when file location object defines FTP or SFTP protocol. 	Source, Target

Related Information

[File format \[page 147\]](#)

[File location object \[page 171\]](#)

[File format properties and modes \[page 147\]](#)

5.3.11.4 Delimiters format property options

The work area on the left, in the [File Format Editor](#) lists file format properties that are not field specific. The Delimiters options are listed in the following table.

Note

See [File format \[page 147\]](#) for mode descriptions.

Delimiters format properties

Option	Possible values	Description	Mode
Column	Tab, Semicolon, Comma, Space, or any character sequence	For delimited file formats, this is the character sequence that indicates the end of one column and the beginning of the next.	New, Edit
Row	{new line}, {Windows new line}, {Unix new line}, or any character sequence	A character sequence that indicates where one row of data ends and the next begins.	New, Edit
Row within text string	Character, Row delimiter	Defines how the row delimiter is interpreted within a text string. <ul style="list-style-type: none">• <i>Character</i>: The specified row delimiter is treated as characters within the text string.• <i>Row delimiter</i>: The specified row delimiter is interpreted and defines rows within the text string.	New, Edit
Text	Single quotation marks ('), double quotation marks ("), or {none}	Denotes the start and end of a text string. All characters (including those specified as column delimiters) between the first and second occurrence of this character is a single text string. The treatment of the row characters is defined by the <i>Row within text string</i> setting.	New, Edit

i Note

Data in columns cannot include the column or row delimiter, unless you also specify a text delimiter. For example, if you specify a comma as your column delimiter, none of the data in the file can contain commas. However, if you specify a comma as the column delimiter and a single quote as the text delimiter, commas are allowed in strings in the data.

You can use any ASCII characters (including non-printing characters) for column and row delimiters.

You can specify an ASCII character by entering a forward slash (/) followed by the decimal representation of the character. For example, to use Y umlaut (Ÿ) as a delimiter, enter /255 in the delimiter property box.

Related Information

[File format \[page 147\]](#)

5.3.11.5 Default Format property options

The work area on the left, in the *File Format Editor* lists file format properties that are not field specific. The *Default Format* property options are listed in the following table.

i Note

See [File format \[page 147\]](#) for mode descriptions.

Default format properties

Option	Possible values	Description	Mode
Escape char	Any character sequence, or {none}	<p>A special character sequence that causes the software to ignore the normal column delimiter. Characters following the escape character sequence are never used as column delimiters.</p> <p>For example, suppose you specify a forward slash as the escape character and a comma as the column delimiter. Then, you must have a forward slash to have a comma appear inside a field.</p>	New, Edit
NULL indicator	{none} or any other character sequence	<p>Special character sequence that the software interprets as NULL data.</p> <p>The software ignores any NULL indicator specified in the file format for blob columns.</p>	New, Edit
Ignore row marker(s)	{none} or any other character sequence	<p>Character sequence, which when found at the beginning of rows, cause the software to ignore the row when reading the file or automatically creating metadata. To enter multiple character sequences, separate each with a semi-colon. To include a semi-colon or backslash as a marking character, precede with a backslash.</p>	New, Edit, Source

Option	Possible values	Description	Mode
Blank padding	leading, trailing	<p>For fixed-width file format targets, pads extra blank spaces before or after the fields.</p> <ul style="list-style-type: none"> • <i>Leading</i>: Adds blanks to the left of (before) the data. • <i>Trailing</i>: Adds blanks to the right of (after) the data. 	New, Edit
Blank trimming	leading, trailing, both, or none	<p>For fixed-width file format sources, trims extra blank spaces before or after the fields.</p> <ul style="list-style-type: none"> • <i>Leading</i>: Trims blanks from the left of (before) the data. • <i>Trailing</i>: Trims blanks from the right of (after) the data. 	New, Edit
Date	yyyy.mm.dd or other combinations	The date format for reading or writing date values to and from the file.	New, Edit
Time	hh24:mi:ss or other combinations	The time format for reading or writing time values to and from the file.	New, Edit
Date-Time	yyyy.mm.dd hh24:mi:ss or other combinations	The datetime format for reading or writing datetime values to and from the file.	New, Edit

Option	Possible values	Description	Mode
Validate decimal data	Yes, No	<p>For file targets, by default, the software converts data in delimited files to the decimal data type (even if it is in string form due to lazy decimal conversion) to make sure that the decimal format is valid.</p> <div> <p>i Note</p> <p>Lazy decimal conversion is an optimization whereby the data for columns of type DECIMAL is stored as STRING data type in rows and is converted to internal DECIMAL format only when they are used in an operation.</p> </div> <p>To improve performance, you can manually deselect this conversion and validation operation. In the Target File Editor, set Validate decimal data to No.</p>	Target

Related Information

[File format \[page 147\]](#)

5.3.11.6 Input/Output format property options

The work area on the left, in the [File Format Editor](#) lists file format properties that are not field specific. The [Input/Output](#) format property options are listed in the following table.

i Note

See [File format \[page 147\]](#) for mode descriptions.

Input/Output format properties

Option	Possible values	Description	Mode
Style	Headers or BOF/EOF	The format of the start and end of the file. Available properties in the Input/Output property group may change, based on this selection.	New, Edit
Skipped rows	Integer	For file formats using Headers style, the number of rows skipped when reading the file. Specify a non-zero value when the file includes comments or other non-data information.	New, Edit
Skip row header	Yes, No	For file formats using Headers style, indicates whether the first row of data in the file contains the column names and should be skipped when reading the file. The software uses this property in addition to the <i>Skipped rows</i> property. When you select <i>Yes</i> , the software does not read data from the first row, and uses data in the first row to determine the file's column names.	New, Edit
Write row header	Yes, No	For file formats using Headers style, indicates whether to write column names in first row of output file.	New, Edit
Write BOM	Yes, No	For file formats using UTF-8 and UTF-16, determines the writing of BOM characters into the file. Choose Yes if you want to include BOM characters into a UTF-8 or UTF-16 file in which byte order is not otherwise defined. For a UTF-16 file, the software assumes the file to be UTF-16be, unless BOM characters are added by this property.	New, Edit
BOF Marker	Any character sequence, including a blank space, an empty string, or {none}	For file formats using BOF/EOF style, the string that marks the start of data in the file.	New, Edit

Option	Possible values	Description	Mode
EOF Marker	Any character sequence, including a blank space, an empty string or {none}	For file formats using BOF/EOF style, the string that marks the end of data in the file.	New, Edit

Related Information

[File format \[page 147\]](#)

5.3.11.7 Locale property options

Set options for Locale in the File Format Editor.

Option	Possible values	Description	Mode
Language	The three-letter language abbreviations specified in the ISO 639-2/T standard	Specifies the human language (for example, Korean, Japanese, or English) in which data is stored or processed. Select from the displayed list.	New, Edit
Code Page	Shows the list of supported code pages. See Supported Locales and Encodings.	Specifies the sequence of bits that defines a character. For example, the Japanese code page contains ASCII, Greek, Cyrillic, and Japanese characters, thereby supporting the English, Greek, Russian, and Japanese languages.	New, Edit

Related Information

[Custom Transfer format property options \[page 163\]](#)

[Locales and Multi-byte Functionality \[page 1378\]](#)

5.3.11.8 Custom Transfer format property options

Descriptions for options in the *Custom transfer* section of the *File Format Editor*.

The work area on the left, in the *File Format Editor* lists file format properties that are not field specific. The *Custom Transfer* format property option descriptions are listed in the following table. These options are available when you select **Yes** for *Custom transfer program* under the *General* options.

→ Tip

Consider using a file location object instead of using a custom transfer program. A file location object allows greater variety of file transfer methods, and provides for editing outside of the data flow.

Custom Transfer format properties

Option	Possible values	Description	Mode
Program executable	File name	(Required) The name of the custom transfer program or its initialization script. For example: MyProgram.exe or MyProgram.cmd.	New, Edit
User name	Any character sequence, including a blank space, an empty string or {none}	Any character sequence, including a blank space, an empty string or {none}	New, Edit
Password	Any character sequence, including a blank space, an empty string or {none}	(optional) Password for the server to which the custom transfer program connects. Passwords entered into this option are encrypted.	New, Edit
Arguments	Any character sequence, including a blank space, an empty string or {none}	(optional) You can create arguments in your custom transfer program and then specifically flag them from within the software using this box. For example, you might have security or compression mechanisms in your program. You can also link connection data to your transfer program's flags.	New, Edit

Related Information

[File format \[page 147\]](#)

[File location object \[page 171\]](#)

5.3.11.9 Locale format property options

The work area on the left, in the *File Format Editor* lists file format properties that are not field specific. The *Local* format property options are listed in the following table.

i Note

See [File format \[page 147\]](#) for mode descriptions.

Locale format properties

Option	Possible values	Description	Mode
<i>Language</i>	The three-letter language abbreviations specified in the ISO 639-2/T standard	Specifies the human language (for example, Korean, Japanese, or English) in which data is stored or processed. Select from the displayed list.	New, Edit
<i>Territory</i>	The two-letter territory abbreviations specified in the ISO 3166-1 standard.	Represents the geographical location (usually the country) where the language is used. The pairing of a language and a territory determines factors such as date format, time format, decimal separator, and so on.	New, Edit
<i>Code page</i>	Shows the list of supported code pages. See Supported Locales and Encodings.	Specifies the sequence of bits that defines a character. For example, the Japanese code page contains ASCII, Greek, Cyrillic, and Japanese characters, thereby supporting the English, Greek, Russian, and Japanese languages.	New, Edit

Related Information

[Locales and Multi-byte Functionality \[page 1378\]](#)

5.3.11.10 Error Handling format property options

The work area on the left, in the *File Format Editor* lists file format properties that are not field specific. The *Error Handling* format property options are listed in the following table.

i Note

See [File format \[page 147\]](#) for mode descriptions.

Error Handling format properties

Option	Possible values	Description	Mode
Log data conversion warnings	Yes, No	Determines whether to include data-type conversion warnings in the error log. Defaults to Yes .	New, Edit, Source
Log row format warnings	Yes, No	Determines whether to include row-format warnings in the error log. Defaults to Yes .	New, Edit, Source
Log warnings	Yes, No	For unstructured file formats, determines whether to log warnings including: <ul style="list-style-type: none"> When there are no files in the specified directory. When no files match the specified filter. When skipping an irregular file on UNIX (for example, a FIFO, symbolic link, character or block device, or UNIX socket). When encountering an empty file with Skip empty files set to Yes. 	New, Edit, Source
Maximum warnings to log	Integer greater than 0 or {no limit}	If Log data conversion warnings or Log row format warnings is enabled, you can limit how many warnings the software logs. Defaults to {no limit} .	New, Edit, Source
Capture data conversion errors	Yes, No	Determines whether to capture data-type conversion errors when processing a flat-file source. Defaults to No .	New, Edit, Source
Capture row format errors	Yes, No	Determines whether to capture row-format errors when processing a flat-file source. Defaults to Yes .	New, Edit, Source
Capture file access errors	Yes, No	For unstructured file formats, determines whether to log file-access errors when processing a flat-file source. Defaults to Yes .	New, Edit, Source
Capture string truncation error	Yes, No	Determines whether to capture string truncation errors when processing a flat-file source. Defaults to No .	New, Edit, Source

Option	Possible values	Description	Mode
Maximum errors to stop job	Integer greater than 0 or {no limit}	If Capture data conversion errors or Capture row format errors is enabled, you can limit how many invalid rows the software processes before stopping the job. Defaults to {no limit} .	New, Edit, Source
Write error rows to file	Yes, No	Determines whether to write invalid rows to an error file. Defaults to No .	New, Edit, Source
Error file root directory	Directory path or blank	<p>If Write error rows to file is enabled, type the root directory in which to save the error file. For added flexibility, you can enter a variable for this option.</p> <p>If you type a directory path here, then only enter the file name in the Error file name property.</p> <p>If you leave Error file root directory blank, then type a full path and file name in the Error file name property.</p>	New, Edit, Source
Error file name	File name, file name including full path name, or blank	<p>If Write error rows to file is enabled, type the name of the file in which to record the invalid rows.</p> <p>For added flexibility, you can enter a variable that is set to a particular file with full path name. Use variables to specify file names that you cannot otherwise enter such as file names that contain multi-byte characters.</p>	New, Edit, Source

Related Information

[File format \[page 147\]](#)

5.3.11.11 Source Information format property options

The work area on the left, in the [File Format Editor](#) lists file format properties that are not field specific. The [Source Information](#) format property options are listed in the following table.

i Note

See [File format \[page 147\]](#) for mode descriptions.

Source Information format properties

Option	Possible values	Description	Mode
Include file name column	Yes, No	<p>Determines whether to add a column that contains the source file name in the source output. Defaults to No.</p> <p>Change the value to Yes when you want to identify the source file in situations such as the following:</p> <ul style="list-style-type: none">You specified a wildcard character to read multiple source files at one time.You load from different source files on different runs.	Source
Column name	N/A	<p>If the file name is included, the name of the column that holds the source file name. Defaults to DI_FILENAME.</p>	Source
Column size	N/A	<p>If the file name is included, the size (in characters) of the column that holds the source file name.</p> <p>Defaults to 100. If the size of the file name column is not large enough to store the file name, truncation occurs from the left.</p>	Source
Include path	N/A	<p>If the file name is included, determines whether to include the full path name of the source file. Defaults to No.</p>	Source

Related Information

[File format \[page 147\]](#)

5.3.11.12 HDFS file format options

File format option descriptions for Hadoop distributed file system (HDFS).

Access the following options in the source or target file editors when you use the HDFS file format in a data flow. Mode refers to creating a new file format, editing a file format, completing source options, or completing target options. The following options appear in all modes.


Option	Possible values	Description	Mode
<i>Data File(s)</i>			
<i>NameNode host</i>	Computer name, fully qualified domain name, IP address, or variable	Name of the NameNode computer. If you use the following default settings, the local Hadoop system uses what is set as the default file system in the Hadoop configuration files. <ul style="list-style-type: none">• <i>NameNode Host</i>: default• <i>NameNode port</i>: 0	All
<i>NameNode port</i>	Positive integer or variable	Port on which the NameNode listens. If you use the following default settings, the local Hadoop system uses what is set as the default file system in the Hadoop configuration files. <ul style="list-style-type: none">• <i>NameNode Host</i>: default• <i>NameNode port</i>: 0	All
<i>Hadoop user</i>	Alphanumeric characters and underscores or variable	Hadoop user name. If you use Kerberos authentication, include the Kerberos realm in the user name. For example: dsuser@BIGDATA.COM .	All

Option	Possible values	Description	Mode
<i>Authentication</i>	Kerberos Kerberos keytab	<p>Indicates the type of authentication for the HDFS connection. Select either value for Hadoop and Hive data sources when they are Kerberos enabled.</p> <p>Kerberos: Select when you have a password to enter in the <i>Password</i> option.</p> <p>Kerberos keytab: Select when you have a generated keytab file. With this option, you do not need to enter a value for <i>Password</i>, but you enter a location for <i>File Location</i>.</p> <p>A Kerberos keytab file contains a list of authorized users for a specific password. The software uses the keytab information instead of the entered password in the <i>Password</i> option. For more information about keytabs, see the MIT Kerberos documentation at http://web.mit.edu/kerberos/krb5-latest/doc/basic/keytab_def.html .</p>	All
<i>File Location</i>	File path	<p>Location for the applicable Kerberos keytab that you generated for this connection.</p> <div> <p>Note</p> <p>This option is only available when you choose Kerberos keytab for the Authentication.</p> </div>	All
<i>Password</i>	Alphanumeric characters and underscores or variable	<p>Password associated with the selected authentication type.</p> <p>This field is required for <i>Authentication</i> type Kerberos. This field is not applicable for <i>Authentication</i> type Kerberos keytab.</p>	All
<i>Root directory</i>	Directory path or variable	Root directory path or variable name for the output file.	All
<i>File name(s)</i>	Alphanumeric characters and underscores or variable	Select the source connection file name or browse to the file by clicking the dropdown arrow. For added flexibility, you can select a variable for this option or use the * wildcard.	All
<i>Pig</i>			
<i>Working directory</i>	Directory path or variable	<p>The Pig script uses this directory to store intermediate data.</p> <div> <p>Note</p> <p>When you leave this option blank, Data Services creates and uses a directory in <code>/user/sapds_temp</code>, within the HDFS.</p> </div>	All

Option	Possible values	Description	Mode
Clean up working directory	Yes, No	<p>Yes: Deletes working directory files</p> <p>No: Preserves working directory files</p> <p>The software stores the Pig output file and other intermediate files in the working directory. Files include scripts, log files, and the <LINK_DIR>/log/hadoop directory.</p> <div> <p>i Note</p> <p>If you select <i>No</i>, intermediate files remain in both the Pig Working Directory and the Data Services directory <LINK_DIR>/log/hadoop.</p> </div>	All
Custom Pig script	Directory path or variable	<p>Location of a custom Pig script.</p> <p>Use the results of the script as a source in a data flow.</p> <p>Custom Pig script can contain any valid Pig Latin command, including calls to any MapReduce jobs that you want to use with Data Services. See your Pig documentation for information about Pig Latin commands.</p> <p>Custom Pig scripts must reside on and be runnable from the local file system that contains the Data Services Job Server that is configured for Hadoop. It is not the Job Server on HDFS. Any external reference or dependency in the script should be available on the Data Services Job Server machine configured for Hadoop.</p> <p>To test your custom Pig script, execute the script from the command prompt and check that it finishes without errors. For example, you could use the following command:</p> <pre>\$ pig -f myscript</pre> <p>Use the results of the Pig script as source in a data flow by using the HDFS file format as a source in a data flow.</p>	All
Locale			
Code page	<default> us-ascii	<p>The applicable Pig code page.</p> <p>The Default option uses UTF-8 for the code page. Select one of these options for better performance.</p> <div> <p>i Note</p> <p>For other types of code pages, Data Services uses HDFS API-based file reading.</p> </div>	All

5.3.12 File location object

Description of a file location object.

Characteristic	Description
	File location icon.
Class	Reusable
Access	Formats tab in the Designer object library.
Description	<p>File location objects contain details about a specific file transfer protocol so that the software safely transfers data from a remote server to a local server (as a source), or from local server to remote server (as a target).</p> <p>Applicable format files are: Flat files, nested schemas (DTD, JSON, XML), COBOL copybooks (source only), and Excel workbooks (source only). Not applicable for source or target messages (DTD, XML).</p>

i Note

You can also set up custom secure file transfer when you create a file format. In the *File Format Editor*, select Yes for *Custom Transfer Program* under the General group of options, and complete the supporting options.

Related Information

[Custom Transfer format property options \[page 163\]](#)

[Designer Guide: File locations, Associate file location objects to file format objects \[page 192\]](#)

[Common options \[page 172\]](#)

[Amazon S3 protocol options \[page 176\]](#)

[Azure Cloud Storage protocol \[page 178\]](#)

[Google Cloud Storage protocol \[page 184\]](#)

5.3.12.1 Common options

Learn about common options for most protocols in the [Create New File Location](#) window.

To open this window, open the [Format](#) tab in the Designer object library, right-click the [File Locations](#) category, and select [New](#).

Some options in the [Create New File Location](#) window are common to more than one protocol type as indicated in the **Type of protocol** column in the following table.

Read about protocol options that are unique to a specific protocol in the associated protocol topics.

Option	Description	Type of protocol
Name	The file name of the file location object.	All types
Protocol	The type of file transfer protocol. The remaining options vary based on the protocol you choose in this option (as noted in the Type of protocol column at right).	All types
Host	The remote server name, fully qualified domain name, or IP address.	FTP, SFTP, SCP
Port	The port number for the remote server. Choose <default> to use the default port for the specified protocol. The fingerprint code for the host computer, expressed in hexadecimal string values.	FTP, SFTP, SCP
Hostkey Fingerprint	The fingerprint code for the host computer, expressed in hexadecimal string values. For information about generating a hostkey fingerprint, see the topic "About the hostkey fingerprint" in this guide.	SFTP, SCP
Authorization Type	Choose either Password or Public Key : <ul style="list-style-type: none">Password: Enter a password for the host in the Password option described later in this table.Public Key: Enter the SSH (UNIX secure shell) authorization information in the SSH Authorization options described later in this table. For information about generating SSH authorization public key information, see the topic "About public key authorization type" in this guide.	FTP, SFTP, SCP
User	The user name for the remote server.	FTP, SFTP, SCP

Option	Description	Type of protocol
<i>Password</i>	The password for the remote server. (For password authorization type only.)	FTP, SFTP, SCP
<i>SSH Authorization Private Key File Path</i>	The file path to the private key file. (For public key authorization type only.)	SFTP, SCP
<i>SSH Authorization Private Key Passphrase</i>	The passphrase for the private key file. (For public key authorization type only.)	SFTP, SCP
<i>SSH Authorization Public Key File Path</i>	The file path to the public key file. (For public key authorization type only.)	SFTP, SCP
<i>Connection Retry Count</i>	The number of times the computer tries to create a connection with the remote server when a previous connection is lost. The default is 10.	All except Local
<i>Connection Retry Interval</i>	The amount of time in seconds that the computer waits until it retries to connect to the remote server when a previous connection is lost. Default is 10 seconds.	FTP, SFTP, SCP
<i>Remote Directory</i>	The file path for the remote server (excluding server name) if applicable. You must have permission to this directory. If you leave this option blank, the software assumes the home directory of your (the user's) file transfer protocol when applicable. When an associated file format is used as a reader (source) in a data flow, the software accesses the remote directory and transfers a copy of the data file to the local directory for processing. When an associated file format is used as a loader (target) in a data flow, the software accesses the local directory location and transfers a copy of the processed file to the remote directory.	FTP, SFTP, SCP, Amazon S3 Cloud Storage

Option	Description	Type of protocol
<i>Local Directory</i>	<p>The file path of the local server you will use for this file location object. The local server directory is located where the Job Server resides. You must have permission to this directory.</p> <div> <p>i Note</p> <p>If this option is blank, the software assumes the directory <code>%DS_COMMON_DIR%/workspace</code> as the default directory.</p> </div> <p>When an associated file format is used as a reader (source) in a data flow, the software accesses the remote directory and transfers a copy of the data file to the local directory for processing.</p> <p>When an associated file format is used as a loader (target) in a data flow, the software accesses the local directory location and transfers a copy of the processed file to the remote directory.</p>	All types

Related Information

[About the hostkey fingerprint \[page 175\]](#)
[About public key authorization type \[page 174\]](#)
[Amazon S3 protocol options \[page 176\]](#)
[Azure Cloud Storage protocol \[page 178\]](#)
[Google Cloud Storage protocol \[page 184\]](#)

5.3.12.1.1 About public key authorization type

When you choose *Public Key* for the *Authorization Type* option in the file location object editor, you must also enter the required SSH information.

Be prepared to supply the following SSH authorization information:

- Private key file path
- Private key passphrase
- Public key file path

Related Information

[Obtain SSH authorization \[page 175\]](#)

5.3.12.1.2 Obtain SSH authorization

The process to obtain the SSH authorization information varies depending on the host system.

1. Execute a command that generates a private key file and a public key file.

For a Linux remote host, the command is: `~/ .ssh`

2. Make the command your current directory.

For Linux remote host, make `~/ .ssh` your current directory.

3. Append the content of the public key file to the `authorized_keys` file.

If you use RSA, the Linux command generates the `id_rsa` private key file and `id_rsa.pub` public key file.

4. Copy the private key file and the public key file to the Job Server host.

If you use RSA, you copy the private key file `id_rsa` and the public key file `id_rsa.pub` to the Job Server host.

5. Enter the private key file name and the public key file name to the options in the file location object.

If you use RSA for Linux (using the example in the steps) you enter the following values to complete the file location object options:

- *SSH Authorized Private Key File Path* = `id_rsa`
- *SSH Authorized Public Key File Path* = `id_rsa.pub`
- *SSH Authorized Private Key Phrase* = Enter a passphrase only if you defined a passphrase when you generated the SSH private key.

5.3.12.1.3 About the hostkey fingerprint

The generated hostkey fingerprint must use MD5 (message digest) or SHA1 (secure hash) algorithm, and the format must be encoded into hexadecimal string values separated by colons.

Hexadecimal string value example: `xx:xx: :xx`. For MD5, the fingerprint length in hexadecimal (including the colon separators) is 47. For SHA1, the fingerprint length in hexadecimal (including the colon separators) is 59.

❖ Example

For a Linux host, obtain the hostkey fingerprint information by first connecting to the file location object host that you defined in the File Location object editor. Then you execute an SSH keygen command such as:

```
user@host> sudo ssh-keygen -lf /etc/ssh/ssh-host_rsa_key.pub
user@host> sudo ssh-keygen -lf /etc/ssh/ssh-host_rsa_key.pub
```

The generated output can be either SHA1 or MD5 format. The generated output is the value that you enter into the *Hostkey Fingerprint* option:

For SHA1 output, you enter the *Hostkey Fingerprint*: 2048 49:fc:79:ef:dd:6c:d3:1b:90:e6:67:9a:d5:93:3a:ac

For MD5 output, you enter the *Hostkey Fingerprint*: root@linux-sles-sp3 (RSA)

5.3.12.2 Amazon S3 protocol options

This topic contains descriptions for options that are unique to the Amazon S3 protocol.

Use a file location object to access data or upload data stored in your Amazon S3 account.

To create a new Amazon S3 file location object:

1. Open the *Format* tab in the Local Object Library in Designer.
2. Right-click the File Location node.
3. Select *New*. The Create New File Location window opens.

The following table describes the file location options that are specific to the Amazon S3 protocol.

Note

You must have "s3:ListBucket" rights in order to view a list of buckets or a special bucket.

Option	Description
<i>Access Key</i>	Amazon S3 identification input value.
<i>Secret Key</i>	Amazon S3 authorization input value.
<i>Region</i>	Name of the region you are transferring data to and from; for example, "South America (Sao Palo)".
<i>Server-Side Encryption</i>	<p>Amazon S3 uses a key to encrypt data at the object level as it writes to disks in the data centers and then decrypts it when the user accesses it.</p> <p>Select <i>None</i>, <i>Amazon S3-Managed Keys</i>, <i>AWS KMS-Managed Keys</i>, or <i>Customer-Provided Keys</i>, depending on the encryption method you want to use.</p> <p>Depending on the encryption choice you make here, none or only one of the three encryption options listed next will be available to you.</p>
<i>Encryption Algorithm</i>	Enter the encryption algorithm that you want to use to encode the data, for example AES256 , aws:kms .

Option	Description
<i>AWS KMS Key ID</i>	You have the option to create and manage encryption keys via the Encryption Keys section in AWS IAM console yourself, or you can leave this field blank to use a default key that is unique to you, the service you're using, and the region you're working in.
<i>AWS KMS Encryption Context</i>	<p>Specify the encryption context of the data. The value is a base64-encoded UTF-8 string holding JSON with the encryption context key-value pairs. For example, if the encryption context is {"fullName": "John Connor" }, you need base64-encoded:</p> <pre>echo {"fullName": "John Connor" } openssl enc -base64 eyJmdWxsTmFtZSI6ICJKb2huIENvbm5veiIgfsANCg==</pre> <p>Enter <pre>eyJmdWxsTmFtZSI6ICJKb2huIENvbm5veiIgfsANCg==</pre> in the encryption context option.</p>
<i>Customer Key</i>	Enter a less-than or equal-to 256 bits key.
<i>Communication Protocol</i>	Communication protocol you are using with S3, either http or https.
<i>Compression Type</i>	<p>The compression type to use.</p> <p>The software compresses the files before upload to S3 and decompresses the files after download from S3.</p>
<i>Connection Retry Count</i>	Number of times the software should try to upload or download data before stopping the upload or download.
<i>Batch size for uploading data, MB</i>	<p>Size of the data transfer you want the software to use for uploading data to S3.</p> <p>Data Services uses single-part uploads for files less than 5 MB in size, and multi part uploads for files larger than 5 MB. Data Services limits the total upload batch size to 100 MB.</p>
<i>Batch size for downloading data, MB</i>	Size of the data transfer the software uses to download data from S3.
<i>Number of threads</i>	Number of upload and download threads for transferring data to S3.
<i>Remote directory</i>	Optional. Name of the directory for Amazon S3 to transfer files to and from.
<i>Bucket</i>	Name of the Amazon S3 bucket containing the data.

Option	Description
<i>Local directory</i>	Optional. Name of the local directory to use to create the files. If you leave this field empty, the software uses the default Data Services workspace.
<i>Proxy host, port, user name, password</i>	Proxy information if you use a proxy server.

Related Information

[Amazon Redshift \[page 70\]](#)

[load_from_s3_to_redshift \[page 1136\]](#)

[Reference Guide: Descriptions of objects, File location object, Common options \[page 172\]](#)

5.3.12.3 Azure Cloud Storage protocol

This topic contains descriptions for options that are unique to the Azure Cloud Storage protocol.

Use a file location object to access data or upload data stored in your Azure Cloud Storage account.

The following table lists the file location object descriptions for the Azure Cloud Storage protocol.

Option	Description
<i>Name</i>	File name of the file location object.
<i>Protocol</i>	Type of file transfer protocol. For Azure, the protocol is Azure Cloud Storage .
<i>Account Name</i>	Name for the Azure storage account in the Azure Portal.
<i>Storage Type</i>	Select Container storage, block blobs. Data Services only supports this type of storage for Azure Cloud Storage.
<i>Authorization Type</i>	Select Primary Shared Key. Data Services only supports this authorization type for Azure Cloud Storage.

Option	Description
<i>Account Shared Key</i>	<p>Copy and paste the primary shared key from the Azure portal in the storage account information.</p> <div> Note <p>For security, the software does not export the account shared key when you export a data flow or file location object that specifies <i>Azure Cloud Storage</i> as the protocol.</p> </div>
<i>Web Service URL</i>	<p>Web services server URL that the data flow uses to access the Web server.</p>
<i>Connection Retry Count</i>	<p>Number of times the computer tries to create a connection with the remote server after a connection fails. After the specified number of retries, Data Services issues an error message and stops the job.</p> <p>The default value is 10. The value cannot be zero.</p>
<i>Batch size for uploading data, MB</i>	<p>Maximum size of a data block per request when transferring data files. The limit is 4 MB.</p> <div> Caution <p>Accept the default setting unless you are an experienced user with an understanding of your network capacities in relation to bandwidth, network traffic, and network speed.</p> </div>
<i>Batch size for downloading data, MB</i>	<p>Maximum size of a data range to be downloaded per request when transferring data files. The limit is 4 MB.</p> <div> Caution <p>Accept the default setting unless you are an experienced user with an understanding of your network capacities in relation to bandwidth, network traffic, and network speed.</p> </div>
<i>Number of threads</i>	<p>Number of upload and download threads for transferring data to Azure Cloud Storage. The default value is 1.</p> <p>When you set this parameter correctly, it could decrease the download and upload time for blobs. For more information, see Number of threads for Azure blobs [page 181].</p>

Option	Description
<i>Remote Path Prefix</i>	<p>Optional. File path for the remote server, excluding the server name. You must have permission to this directory.</p> <p>If you leave this option blank, the software assumes that the remote path prefix is the user home directory used for FTP.</p> <p>When an associated file format is used as a reader in a data flow, the software accesses the remote directory and transfers a copy of the data file to the local directory for processing.</p> <p>When an associated file format is used as a loader in a data flow, the software accesses the local directory location and transfers a copy of the processed file to the remote directory</p> <p>Container type storage is a flat file storage system and it does not support subfolders. However, Microsoft allows forward slashes with names to form the remote path prefix, and a virtual folder in the container where you upload the files.</p> <div> <p>❖ Example</p> <p>You currently have a container for finance database files. You want to create a virtual folder for each year. For 2016, you set the remote path prefix to: 2016/. When you use this file location, all of the files upload into the virtual folder "2016".</p> </div>
<i>Local Directory</i>	<p>Path of your local server directory for the file upload or download.</p> <p>Requirements for local server:</p> <ul style="list-style-type: none"> • must exist • located where the Job Server resides • you have appropriate permissions for this directory <p>When an associated file format is used as a reader in a data flow, the software accesses the remote directory and transfers a copy of the data file to the local directory for processing.</p> <p>When an associated file format is used as a loader in a data flow, the software accesses the local directory location and transfers a copy of the processed file to the remote directory.</p>

Option	Description
Container	<p>Azure container name for uploading or downloading blobs to your local directory.</p> <p>If you specified the connection information, including account name, shared key, and proxy information (if applicable), click the Container field. The software sends a request to the server for a list of existing containers for the specific account. Either select an existing container or specify a new one. When you specify a new one, the software creates it when you run a job using this file location object.</p>
Proxy Host, Port, User Name, Password	Optional. Enter the proxy information if you use a proxy server.

Related Information

[Common options \[page 172\]](#)

[File location object \[page 171\]](#)

5.3.12.3.1 Number of threads for Azure blobs

The number of threads is the number of parallel uploaders or downloaders to be run simultaneously when you upload or download blobs.

The [Number of threads](#) setting affects the efficiency of downloading and uploading blobs to or from Azure Cloud Storage.

Determine the number of threads

To determine the number of threads to set for the Azure file location object, base the number of threads on the number of logical cores in the processor that you use.

Example thread settings

Processor logical cores	Set Number of threads
8	8
16	16

The software automatically re-adjusts the number of threads based on the blob size you are uploading or downloading. For example, when you upload or download a small file, the software may adjust to use fewer

numbers of threads and use the block or range size you specified in the [Batch size for uploading data, MB](#) or [Batch size for downloading data, MB](#) options.

Upload Blob to an Azure container

When you upload a large file to an Azure container, the software may divide the file into the same number of lists of blocks as the setting you have for [Number of threads](#) in the file location object. For example, when the [Number of threads](#) is set to 16 for a large file upload, the software divides the file into 16 lists of blocks. Additionally, each thread reads the blocks simultaneously from the local file and also uploads the blocks simultaneously to the Azure container.

When all the blocks are successfully uploaded, the software sends a list of commit blocks to the Azure Blob Service to commit the new blob.

If there is an upload failure, the software issues an error message. If they already existed before the upload failure, the blobs in the Azure container stay intact.

When you set the number of threads correctly, you may see a decrease in upload time for large files.

Download Blob from an Azure container

When you download a large file from the Azure container to your local storage, the software may divide the file into the [Number of threads](#) setting in the file location object. For example, when the [Number of threads](#) is set to 16 for a large file download to your local container, the software divides the blobs into 16 lists of ranges. Additionally, each thread downloads the ranges simultaneously from the Azure container and also writes the ranges simultaneously to your local storage.

When your software downloads a blob from an Azure container, it creates a temporary file to hold all of the threads. When all of the ranges are successfully downloaded, the software deletes the existing file from your local storage if it existed, and renames the temporary file using the name of the file that was deleted from local storage.

If there is a download failure, the software issues an error message. The existing data in local storage stays intact if it existed before the download failure.

When you set the number of threads correctly, you may see a decrease in download time.

5.3.12.4 Azure Data Lake Store protocol options

Use an Azure Data Lake Store file location object to read data from and upload data to your Azure Data Lake Store.

When you create the file location object, select [Azure Data Lake Store](#) from the [Protocol](#) dropdown list.

The following table describes the file location options that are specific to the Azure data lake Store protocol.

Option	Description
<i>Data Lake Store name</i>	<p>Name of the Azure Data Lake Store to access.</p> <p>Optionally use a substitution parameter.</p>
<i>Service Principal ID</i>	Obtain from your Azure Data Lake Store administrator.
<i>Tenant ID</i>	Obtain from your Azure Data Lake Store administrator.
<i>Password</i>	Obtain from your Azure Data Lake Store administrator.
<i>Batch size for uploading data (MB)</i>	<p>Maximum size of a data block to upload per request when transferring data files. The default setting is 5 MB.</p> <div> <p>⚠ Caution</p> <p>Keep the default setting unless you are an experienced user with an understanding of your network capacities in relation to bandwidth, network traffic, and network speed.</p> </div>
<i>Batch size for downloading data (MB)</i>	<p>Maximum size of a data range to download per request when transferring data files. The default setting is 5 MB.</p> <div> <p>⚠ Caution</p> <p>Keep the default setting unless you are an experienced user with an understanding of your network capacities in relation to bandwidth, network traffic, and network speed.</p> </div>
<i>Number of threads</i>	Number of parallel uploaders or downloaders to run simultaneously. The default value is 1.
<i>Remote path prefix</i>	<p>Directory path for your files in the Azure Data Lake Store. Obtain the directory path from Azure Data Lake Store Properties.</p> <div> <p>❖ Example</p> <p>If the the directory in your Azure Data Lake Store Properties is <code>adl://</code></p> <p><code><yourdatastoreName>.azuredatalakestore.net/<FolderName>/<subFolderName></code>, the remote path prefix value is <code><FolderName>/<subFolderName></code>.</p> </div> <p>Permission to access this directory required.</p> <p>Optionally use substitution parameter.</p>

Option	Description
Local directory	<p>Path to the local directory for your local Data Lake Store data.</p> <p>Permission to access this directory required.</p> <p>Optionally use substitution parameter.</p>

Related Information

[File location object \[page 171\]](#)

[Common options \[page 172\]](#)

5.3.12.5 Google Cloud Storage protocol

This topic contains descriptions for options that are unique to the Google cloud Storage protocol.

Use a file location object to access data or upload data stored in your Google Cloud Storage account.

The following table lists the file location object descriptions for the Google Cloud Storage protocol.

Option	Description
Name	File name of the file location object.
Protocol	<p>Type of file transfer protocol.</p> <p>For Google, the protocol is Google Cloud Storage.</p>
Project	Google BigQuery project name.
Upload URL	Accept the default, https://www.googleapis.com/upload/storage/v1 .
Download URL	Accept the default, https://www.googleapis.com/storage/v1 .
Authentication Server URL	<p>Accept the default, https://accounts.google.com/o/oauth2/token.</p> <p>The default is the Google URL plus the name of the Web access service provider, OAuth 2.0.</p>

Option	Description
<i>Authentication Access Scope</i>	<p>Enables access to specific user data. <i>Cloud-platform</i> is the default.</p> <ul style="list-style-type: none"> • <i>Read-only</i>: Allows access to read data, including listing buckets. Google information about read-only: https://www.googleapis.com/auth/devstorage.read_only ➤ • <i>Read-write</i>: Allows access to read and change data, but not metadata like ACLs. Google information about read-write: https://www.googleapis.com/auth/devstorage.read_write ➤ • <i>Full-control</i>: Allows full control over data, including the ability to modify ACLs. Google information about full-control: https://www.googleapis.com/auth/devstorage.full_control ➤ • <i>Cloud-platform.read-only</i>: View your data across Google Cloud Platform services. For Google Cloud Storage, this option is the same as devstorage.read-only. Google information about cloud-platform.read-only: https://www.googleapis.com/auth/cloud-platform.read-only ➤ • <i>Cloud-platform</i>: View and manage data across all Google Cloud Platform services. For Google Cloud Storage, this option is the same as devstorage.full-control. Google information about cloud-platform: https://www.googleapis.com/auth/cloud-platform ➤
<i>Service Account Email Address</i>	Enter the e-mail address from your Google project. This e-mail is the same as the service account e-mail address that you enter into the applicable Google BigQuery datastore.
<i>Service Account Private Key</i>	Click the <i>Browse</i> icon and select the .p12 file that you created in your Google project and downloaded locally. Click <i>Open</i> .
<i>Service Account Signature Algorithm</i>	Accept the default: SHA256withRSA. This value is the algorithm type that the software uses to sign JSON Web Tokens. The software uses this value, along with your service account private key, to obtain an access token from the Authentication Server.
<i>Substitute Access Email Address</i>	Optional. Enter the substitute e-mail address from your Google BigQuery application datastore.
<i>Web Service URL</i>	Web services server URL that the data flow uses to access the Web server.
<i>Compression Type</i>	Select <i>None</i> or <i>gzip</i> . The gzip type lets you upload gzip files to Google Cloud Storage.

Option	Description
<i>Connection Retry Count</i>	<p>Number of times the computer tries to create a connection with the remote server after a connection fails. After the specified number of retries, Data Services issues an error notification and stops the job.</p> <p>The default value is 10. The value cannot be zero.</p>
<i>Batch size for uploading data, MB</i>	Maximum size of a data block to be uploaded per request when transferring data files. The limit is 5 TB.
<i>Batch size for downloading data, MB</i>	Maximum size of a data block to be downloaded per request when transferring data files. The limit is 5 TB.
<i>Number of threads</i>	<p>Number of upload and download threads for transferring data to Google Cloud Storage.</p> <p>The default is 1.</p> <p>Enter a number from 1 to 30. If you enter any number outside this range, the software automatically adjusts the number at runtime.</p>
<i>Bucket</i>	<p>Bucket name, which is the name of the basic container that holds your data.</p> <p>Select a bucket name from the dropdown list. The list only contains bucket names that exist in the datastore. To create a new bucket, enter the name of the bucket here. If the bucket does not exist in Google Cloud Storage, Google creates the bucket when you perform an upload for the specified bucket.</p> <div> <p>i Note</p> <p>If you attempt to download the bucket and it does not exist in Google, the software issues an error.</p> </div>
<i>Remote Path Prefix</i>	<p>Optional. Folder structure of the Google Cloud Storage bucket. It should end with a forward slash (/). For example, <code>test_folder1/folder2/</code>. You must have permission to this directory.</p> <p>If you leave this option blank, the software assumes the home directory of your file transfer protocol.</p> <p>When an associated file format is used as a reader in a data flow, the software accesses the remote directory and transfers a copy of the data file to the local directory for processing.</p> <p>When an associated file format is used as a loader in a data flow, the software accesses the local directory location and transfers a copy of the processed file to the remote directory</p>

Option	Description
Local Directory	<p>The file path of the local server that you use for this file location object. The local server directory is located where the Job Server resides. You must have permission to this directory.</p> <div> <p>i Note</p> <p>If this option is blank, the software assumes the directory %DS_COMMON_DIR%/workspace as the default directory.</p> </div> <p>When an associated file format is used as a reader in a data flow, the software accesses the remote directory and transfers a copy of the data file to the local directory for processing.</p> <p>When an associated file format is used as a loader in a data flow, the software accesses the local directory location and transfers a copy of the processed file to the remote directory.</p>
Proxy Host, Port, User Name, Password	Optional. Enter the proxy information if you use a proxy server.

5.3.12.6 HDFS file location object options

Use a Hadoop distributed file system (HDFS) file location to access your Hadoop data for Data Services processing.

Use the HDFS file location as a source or target in a SAP Data Services data flow.

When you create a new HDFS file location, select [HDFS](#) from the [Protocol](#) dropdown list.

The following table describes the file location options that are specific to the HDFS protocol. For descriptions of general options, see the *Reference Guide*.

Option	Description
Connection section	
Protocol	<p>Type of file transfer protocol.</p> <p>Select HDFS.</p>

Option	Description
<i>Communication Protocol</i>	<p>Type of protocol to use to access the data in your HDFS.</p> <ul style="list-style-type: none"> • <i>WebHDFS</i>: Select when Data Services is not installed as a part of the Hadoop cluster. Ensure that you configure WebHDFS on your server side. • <i>HTTPS</i>: Select when Data Services is not installed as a part of the Hadoop cluster. • <i>HDFS</i>: Select when Data Services is installed as a part of the Hadoop cluster.
<i>Host</i>	Name of the computer that hosts the NameNode.
<i>Secondary NameNode</i>	Name of the computer that hosts the secondary NameNode.
<i>Port</i>	Port number on which the NameNode listens.
<i>User</i>	Hadoop user name.
<i>Password</i>	Password for the WebHDFS communication protocol method.
<i>Compression type</i>	<p>Specifies to not use compression, or to use gzip compression:</p> <ul style="list-style-type: none"> • <i>None</i>: Default setting. The file location object does not use compression. • <i>gzip</i>: The file location object uses gzip compression. The software compresses the files before upload to Hadoop and decompresses the files after download from Hadoop. <p>Not applicable when you select <i>HDFS</i> type for <i>Communication protocol</i>.</p>
<i>Connection retry count</i>	<p>Number of times the computer tries to create a connection with the remote server after a connection fails.</p> <p>The default value is <i>10</i>.</p> <p>The value cannot be zero.</p> <p>After the specified number of retries, Data Services issues an error message and stops the job.</p> <p>Not applicable when you select <i>HDFS</i> type for <i>Communication protocol</i>.</p>

Option	Description
<i>Batch size for uploading data (MB)</i>	<p>Size of the data transfer in MB to use for uploading data.</p> <p>The default value is <i>5</i>.</p> <p>Data Services uses different upload methods based on file size:</p> <ul style="list-style-type: none"> • Single part uploads for files less than 5 MB. • Multi part uploads for files larger than 5 MB. <p>Data Services limits the total upload batch size to 100 MB.</p> <p>Not applicable when you select <i>HDFS</i> type for <i>Communication protocol</i>.</p>
<i>Batch size for downloading data MB</i>	<p>Size of the data transfer in MB the software uses to download data from Hadoop.</p> <p>The default value is <i>5</i>.</p> <p>Not applicable when you select <i>HDFS</i> type <i>Communication protocol</i>.</p>
<i>Number of threads</i>	<p>Number of upload and download threads for transferring data from and to Hadoop.</p> <p>The default value is <i>1</i>.</p> <p>Not applicable when you select <i>HDFS</i> type <i>Communication protocol</i>.</p>
<i>Authentication type</i>	<p>Authentication for the HDFS connection.</p> <ul style="list-style-type: none"> • <i>None</i>: Kerberos security is not enabled. <p>For Kerberos enabled cluster:</p> <ul style="list-style-type: none"> • <i>Delegation token</i>: You have a delegation token for authentication of the request. • <i>Kerberos</i>: Default. You have a password to enter in the <i>Password</i> option. • <i>Kerberos keytab</i>: You have a generated keytab file. With this option, you do not enter a value for <i>Password</i>, but you enter a location for <i>Keytab file</i>.
<i>Keytab file</i>	<p>Generated keytab file name.</p> <p>Applicable when you select <i>Kerberos keytab</i> for <i>Authentication type</i>.</p>
<i>Kerberos Password</i>	<p>Kerberos password.</p> <p>Applicable when you select <i>Kerberos</i> for <i>Authentication type</i>.</p>

Option	Description
<i>SSL enabled</i>	<p>Select Yes to use a Secure Socket Layer (SSL) connection to HDFS.</p> <p>Not applicable when you select WebHDFS type for Communication protocol.</p>
<i>File System</i> section	
<i>Remote directory</i>	Path for your HDFS working directory.
<i>Local directory</i>	Path for your local working directory.
<i>Replication factor</i>	<p>The number of replicated files that HDFS should create.</p> <p>The default value is 2.</p> <p>Not applicable when you select HDFS type for Communication protocol.</p>
<i>Proxy</i> section: Complete the Proxy options only when you are using a proxy.	
<i>Proxy host</i>	<p>Path and host name for the REST API proxy server.</p> <p>Not applicable when you select HDFS type for Communication protocol.</p>
<i>Proxy port</i>	<p>Port number for the REST API Proxy server.</p> <p>Not applicable when you select HDFS type for Communication protocol.</p>
<i>Proxy Username</i>	<p>User name for the REST API proxy server.</p> <p>Not applicable when you select HDFS type for Communication protocol.</p>
<i>Proxy Password</i>	<p>Password for the REST API proxy server.</p> <p>Not applicable when you select HDFS type for Communication protocol.</p>
<i>HDFS Proxy user</i>	<p>Proxy user name configured for the HDFS user.</p> <p>Not applicable when you select HDFS type for Communication protocol.</p>
<i>Pig</i> section	

Option	Description
Working directory	<p>Directory path or variable. The software uses this directory when transferring data from the remote server to the local server, and when transferring data from the local server to the remote server.</p> <p>Applicable only for <i>HDFS</i> type <i>Communication protocol</i>.</p>
Clean up working directory	<p>Determines if the software deletes files in the working directory after execution.</p> <ul style="list-style-type: none"> • Yes: Default setting. Deletes the working directory files. • No: Preserves the working directory files. <p>If you select No, intermediate files remain in both this working directory and the Data Services directory < \$LINK_DIR>/log/hadoop.</p> <p>Applicable only for <i>HDFS</i> type <i>Communication protocol</i>.</p>
Custom Pig script	<p>Directory path or variable. Location of a custom Pig script, if applicable.</p> <p>Applicable only for <i>HDFS</i> type <i>Communication protocol</i>.</p>

Related Information

[Common options \[page 172\]](#)

[File location object \[page 171\]](#)

[Prerequisites to Data Services configuration \[page 1432\]](#)

[Setting up HDFS and Hive on Windows \[page 1436\]](#)

[Connect to HDFS \[page 1437\]](#)

5.3.12.7 Add multiple protocol types

You can create multiple configurations for one file location object and easily switch protocols for different jobs without changing the data source object in the data flow.

Control the type of protocol to use in the file location object by changing the default configuration in the file location object editor.

For example, create one file location object named `all_protocols` for FTP. Create additional configurations in the `all_protocols` file location object for SFTP, SCP, and Local. Process the data flow in your test environment using the default configuration, FTP. Then change the default configuration to SFTP in the `all_protocols` file location object and rerun the same job. Unless you have other properties to change in the data flow, you don't need to open the dataflow to switch the protocol settings.

5.3.12.8 Associate file location objects to file formats

To use the file transfer protocol information that you entered into a file location object, you associate the file location object with a specific file source or target in a data flow.

You can also include a file location object when you create or edit a new format.

i Note

Some formats are not applicable for file location objects. In those cases, the file location object options don't apply. See the following table for format types.

The following table provides an overview of how to associate a specific file location object with a specific format type.

Format type	Associate file location object
Flat file	Create a new or edit a flat file format, and add the file location object information in the Data Files section of the File Format Editor .
DTD, XML, or JSON schema file	Edit a schema file and add the file location object information in the Format tab of the Format Editor .
COBOL copybook	Edit an existing COBOL copybook and enter the file location information in the Data File tab of the Edit COBOL Copybook dialog.
Excel workbook	Edit an existing Excel workbook and enter the file location information in the Format tab of the Import Excel Workbook dialog.

5.3.12.9 Use built-in functions for file transfer

Use file transfer built-in functions in scripts in a work flow to incorporate a file location object's information for file transfer protocol.

You can use the built-in functions `copy_to_remote_system` and `copy_from_remote_system` in scripts in your work flow to move files from local to remote server or to move files from remote to local server. You can use the scripts alone or in combination with a file format as source or target.


Related Information

[copy_from_remote_system](#) [page 1058]

[copy_to_remote_system](#) [page 1060]

[File location object](#) [page 171]

5.3.13 Function

Characteristic	Description
	Function icon.
Class	Reusable
Access	<ul style="list-style-type: none"> For existing functions, click the Functions button in object editors. For imported functions, in the object library, click the Datastores tab, expand a datastore, and expand the Functions node. For custom or validation functions, click the Custom Functions tab in the object library or select Tools > Custom Functions.
Description	<p>Use functions to process values. There are several types of functions:</p> <ul style="list-style-type: none"> Built-in functions DBMS and application functions or stored procedures imported into SAP Data Services Custom functions you create Validation functions that you can import from SAP Information Steward or create locally

Related Information


5.3.13.1 Function attributes

Functions have the following common attributes:

Attribute	Description
Name	The name of the function. This name appears in the function wizard and smart editor. It is also used when the function appears in a script or expression.
Description	Descriptive text entered when the function is created or imported into the software.
Function type	<p>Each imported or custom function has a type. For imported and custom functions, right-click the function from the object library and select Properties to view the type. Descriptions and syntax for built-in functions is listed in the function wizard and the smart editor.</p> <p>Some functions also include a Category designation on the Function tab of the Properties dialog box.</p>

Attribute	Description
Enable Parallel Execution	<p>Check box on the Properties dialog box. Enables the software to run stored procedures and custom functions in parallel.</p> <p>This option must be selected in addition to entering a positive number for the parent data flow's degree of parallelism.</p> <p>For more information, see "Degree of parallelism" in the <i>Performance Optimization Guide</i>.</p>
Validation function	Check box on the Properties dialog box indicating if the function is a validation function.

5.3.14 JSON

Characteristic	Description
	JSON icon.
Class	Reusable
Access	In the object library, click the <i>Formats</i> tab, then open the Nested Schemas category.
Description	<p>A JSON describes the data schema of a JSON message or file.</p> <p>Data flows can read and write data to messages or files based on a specified JSON format. You can use the same JSON to describe multiple JSON sources or targets.</p> <p>To use JSONs, import metadata into SAP Data Services. During import, the software converts the structure defined in the JSON into the nested-relational data model (NRDM).</p>

5.3.14.1 Editor

Open the JSON editor by double-clicking a JSON name in the object library.

5.3.14.2 JSON Properties

Context-click a JSON nested schema in the *Formats* tab of the object library and select *Properties*.

There are two tabs in the JSON file properties dialog: *General* and *Format*.

General tab

Property	Description
Name	(Read only) The name of the format. This name appears in the object library under the Nested Schemas category of the Formats tab and is used for sources and targets (JSON files or messages) that reference this format in data flows.
Description	(Read-only) The description for the JSON, if applicable.

Format tab

Option	Description
File Location	<p>(Optional) Select the name of an existing file location object.</p> <p>The file location object contains file transfer protocol information and local and remote server information to safely transfer data from remote to local server (source) and local to remote server (target).</p> <div> <p>i Note</p> <p>When this option is enabled, the Directory option is disabled, and the Data Access tab is also disabled.</p> </div>
Delete file after transfer	<p>Available when you select a file location object above.</p> <p>As a source:</p> <ul style="list-style-type: none"> Yes: Not applicable for SCP file transfer protocol. Deletes local file after software reads data from local file into data flow. No: Saves local file after software reads data from local file into data flow. <p>As a target:</p> <ul style="list-style-type: none"> Yes: Overwrites the target local file with generated output data. No: Appends the target local file with generated output data.
File name	Select the JSON file name or browse to the JSON file (.json extension) by clicking the drop list arrow. For added flexibility, you can select a variable for this option.
Imported from	(Read only) Contains the full path to the JSON file.

5.3.15 Log

Characteristic	Description
Class	Single-use
Access	<ul style="list-style-type: none">To see the logs for jobs run on a particular Job Server, log in to the repository associated with the Job Server when you open the Designer. In the project area of the Designer, click the Log tab, and expand the job tree.To see the logs for jobs run on a particular Job Server, in the Administrator, select Batch Jobs > Repository (selecting the repository associated with the Job Server). Then, in the Job Information column for a job execution, click the type of log you want to view.
Description	<p>A log records information about a particular execution of a single job.</p> <ul style="list-style-type: none">The Log tab in the Designer displays all logs for each execution. When you are finished with the logs for a given job or project, delete them from the Log tab. Right-click the log and select Delete Log.The Job Information column, of the Batch Job Status page in the Administrator also displays all logs for each execution. <p>There are three types of logs:</p> <ul style="list-style-type: none">Trace logsMonitor logsError logs

Related Information

[Trace logs \[page 196\]](#)

[Monitor logs \[page 198\]](#)

[Error logs \[page 199\]](#)

5.3.15.1 Trace logs

The tracelog shows the execution progress through each component (object) of the job. It lists the process ID, thread ID, the object type being executed, the time each event began, and a description of the event.

For unsuccessful jobs, use the trace log to see which components of a partially executed job completed or where an error occurred.

Trace logs have the following information:

Entry	Description
Pid	Indicates the process identification number of the thread executing.

Entry	Description
Tid	Indicates the thread identification number of the thread executing.
Type	Indicates the object being executed, such as a data flow or a transform. The generic job events are labeled TRACE. Possible types are listed and described in the following table.
TimeStamp	Indicates the date and time when the thread generated the message.
Message	Gives a description of the event that occurred as the thread was executing.

There are several types of traces.

Error number prefix	Description
ABAP	Traces the ABAP query execution.
ADMIN	Prints administrative information like "server not responding" or "power failure."
BLKLOAD	Traces bulk loading.
DATAFLOW	Traces the data flow execution.
EMAIL	Traces e-mail messages.
FTP	Traces FTP transport.
JOB	Traces the job execution.
OPTIMIZE	Records optimized details.
REPO	Traces objects in the repository.
ROW	Traces the row as it passes from one transform to another. It prints the row that is input to the transform and the output row it generates.
SQLFUNC	Traces function execution.
SQLLOAD	Traces loader execution, including the SQL sent to the target database.
SQLREAD	Traces reader execution, including the SQL sent to the source database.
SQLTRAN	Traces SQL transforms such as Table_Comparison and Key_Generation. The trace includes the SQL query sent to the underlying database and SQL results returned.
TRAN	Traces the transform execution.
USERFUNC	Traces user functions.

5.3.15.2 Monitor logs

The monitor log quantifies the activities of the components of the job. It lists the time spent in a given component of a job and the number of data rows which streamed through the component.

Use the monitor log to help tune the performance of a job.

Monitor logs have the following information:

Entry	Description
Path Name	<p>Indicates which object is executing. The path name has the following format:</p> <pre><dfname[_subdataflownumber]/objectname ></pre> <p>where</p> <ul style="list-style-type: none">• <code><dfname></code> is the name of the data flow• <code><_subdataflownumber></code> is the number of the sub data flow if SAP Data Services split the data flow into multiple sub data flows• <code><objectname ></code> is the name of the source, transform, or target that the data flow is processing <p>For example, the following path name is for the first sub data flow of a data flow named Orders_DF, and the object being processed is a source named Order Details:</p> <pre>/Orders_DF_1/ORDER DETAILS</pre> <p>The next example is a path name for the first sub data flow of the Orders_DF data flow, and the object being processed is temporary storage for data from Order Details. The 'TS' indicates that this object is temporary storage, and 'ORDERTEMP' is the name of the temporary storage specified in the Data_Transfer transform.</p> <pre>/Orders_DF_1/TS_ORDER DETAILS_ORDERTEMP</pre> <p>The next example is a path name for the second sub data flow of the Orders_DF data flow, and the object being processed is a query transform:</p> <pre>/Orders_DF_2/Query</pre>
State	<p>Indicates the current status of the execution of the object. If you view the log while the job is running, this value changes as the status changes. The possible values are START, PROCEED, and STOP. In a successfully run job, all of these values are STOP to indicate that they finished successfully.</p>
Row Count	<p>Indicates the number of rows processed through this object. This value is updated based on the Monitor sample rate (# of seconds) set as a debug property.</p>
Elapsed Time	<p>Indicates the time (in seconds) since this object received its first row of data.</p>
Absolute Time	<p>Indicates the time (in seconds) since the execution of this entire data flow (including all of the transforms) began.</p>

5.3.15.3 Error logs

The [error log](#) lists errors generated by SAP Data Services, by the source or target DBMS, or the operating system. If the error log is empty (that is, the button is dimmed), the job completed successfully.


Error logs have the following information:

Entry	Description
Pid	The process thread identification number of the thread executing.
Tid	The thread identification number of the thread executing.
Number	An error number prefix (explained in the following table) and a number.
TimeStamp	The date and time when the thread generated the message.
Message	A description of the error that occurred as the thread was executing.


The error number prefixes are as follows:

Error number prefix	Description
ADM	Administration errors.
BAP	BAPI errors.
BIW	SAP BW errors.
CON	Connection errors. The connection indicated could not be initialized or failed during execution.
DBS	Database management system errors.
EML	Email errors.
FIL	Filespec errors.
OPT	Optimization errors.
PAR	Parser errors.
R3C	SAP connectivity errors.
R3S	SAP syntax errors.
REP	Repository errors.
RES	Resolver errors.
RUN	Runtime errors.
SCH	Job launcher errors.
SRV	Job Server errors.
SYS	System exceptions.
USR	User function errors.
VAL	Validator errors.
XRN	Transform errors.

5.3.16 Message function

Characteristic	Description
	Message function icon.
Class	Reusable
Access	In the object library, click the Datastores tab.
Description	<p>Available in certain adapter datastores, message functions can accommodate XML messages when properly configured.</p> <p>See your adapter's documentation for more specific information about the options available for a message function.</p>

5.3.17 Nested Schemas template

Characteristic	Description						
	Nested schemas template icon.						
Class	Single-use						
Access	<table><tr><th>Scenario</th><th>Description</th></tr><tr><td>Inserting as a target</td><td>Select the Nested Schemas Template icon in the tool palette, then click the data flow diagram in the workspace.</td></tr><tr><td>Viewing options</td><td>Click the name of the Nested Schemas template in the workspace or in the project area. This opens the object editor.</td></tr></table>	Scenario	Description	Inserting as a target	Select the Nested Schemas Template icon in the tool palette, then click the data flow diagram in the workspace.	Viewing options	Click the name of the Nested Schemas template in the workspace or in the project area. This opens the object editor.
Scenario	Description						
Inserting as a target	Select the Nested Schemas Template icon in the tool palette, then click the data flow diagram in the workspace.						
Viewing options	Click the name of the Nested Schemas template in the workspace or in the project area. This opens the object editor.						

Characteristic	Description
Description	<p>Use a Nested Schemas template to create a JSON file or an XML file that matches a particular input schema. The Nested Schemas template does not require and does not produce a corresponding JSON Schema, XML Schema, or DTD format. Likewise when it generates XML, it does not create column attributes if they are present in its input schema.</p> <p>Therefore, you may use the Nested Schemas template to produce:</p> <ul style="list-style-type: none"> • JSON file without predefining a JSON format • XML file without predefining an XML format <p>You can use a Nested Schemas template as a target in a batch or real-time job.</p> <div> <p>i Note</p> <p>In a Nested Schemas template, all data types are converted to varchar.</p> </div> <p>After adding a Nested Schemas template to a data flow, specify the name and location of the file. In the Nested Schemas template target file editor, select either the <i>XML</i> or the <i>JSON</i> option as applicable.</p> <div> <p>i Note</p> <p>When using Nested Schemas templates in real-time jobs, deselect the <i>Delete and recreate file</i> option in the target editor. This option is selected by default when you create a Nested Schemas target.</p> </div>

Related Information


[Target XML files, messages, and templates \[page 271\]](#)

5.3.18 Outbound message


Characteristic	Description
	Outbound message icon.
Class	Reusable
Access	In the object library, click the <i>Datastores</i> tab.

Characteristic	Description
Description	<p>Available in some adapter datastores, outbound messages are XML-based, hierarchical communications that SAP Data Services can publish to adapters. Outbound messages only wait for an acknowledgement from an external system; they do not wait for a reply. You can use outbound messages as targets only. You cannot use outbound messages as sources.</p> <p>See your adapter's documentation for more specific information about the options available for outbound messages.</p>

5.3.19 Project


Characteristic	Description				
	Project icon.				
Class	Single-use				
Access	<ul style="list-style-type: none"> Choose ► Project ► New ►. In the object library, click the Projects tab. 				
Description	<p>A project allows you to group jobs. It is the highest level of organization offered by SAP Data Services. Opening a project makes one group of jobs easily accessible in the user interface.</p> <p>Projects have the following attribute:</p> <table> <tr> <th>Attribute</th><th>Description</th></tr> <tr> <td>Name</td><td>The name of the object, which appears on the object in the project area.</td></tr> </table>	Attribute	Description	Name	The name of the object, which appears on the object in the project area.
Attribute	Description				
Name	The name of the object, which appears on the object in the project area.				

5.3.20 Query transform

Characteristic	Description
	Query icon.
Class	Single-use
Access	With a data flow diagram in the work space, click the Query transform icon in the tool palette, then click in the work space.

Characteristic	Description				
Description	<p>A Query transform, like a SQL SELECT statement, retrieves a data set that satisfies the conditions you specify. With a Query transform, you can:</p> <ul style="list-style-type: none"> • Map columns from input to output schema • Add new columns, nested schemas, and functions to the output schema • Choose the data to extract • Perform operations on the data • Join data from multiple sources <p>A Query transform can operate on nested data. Using a Query transform, you can nest data or unnest nested data.</p> <p>Query transforms have one attribute:</p> <table> <tr> <th>Attribute</th><th>Description</th></tr> <tr> <td>Name</td><td>The name of the object, which appears on the object in the diagram.</td></tr> </table>	Attribute	Description	Name	The name of the object, which appears on the object in the diagram.
Attribute	Description				
Name	The name of the object, which appears on the object in the diagram.				

5.3.21 Real-time job

Characteristic	Description
	Real time job icon.
Class	Reusable
Access	<ul style="list-style-type: none"> • In the object library, click the Jobs tab. • In the project area, right-click a project and select Real-time Job.

Characteristic	Description
Description	<p>A real-time job is a set of objects that you can execute together to process messages.</p> <p>A real-time job is made up of three logical components:</p> <ul style="list-style-type: none"> • Initialization (optional) • Real-time processing loop • Clean-up (optional) <p>Each component can include the same objects as a batch job.</p> <p>A real-time job is created in the Designer and then configured in the Administrator as a real-time service associated with an Access Server.</p> <p>Start real-time services in the Administrator. If you have included any objects in the initialization component of a real-time job, they run when the service starts. When a real-time service starts, a real-time processing loop registers itself and its message type with the Access Server and waits for the Access Server to send requests. The real-time processing loop continues to run until it encounters an error or you shut it down using the Administrator. The objects you placed inside the clean-up component of a real-time job run only when a service is shut down.</p> <p>The message type that a given real-time job processes is determined (when it is designed) by the real-time source you include in the real-time processing loop; the format of the response is determined by the real-time target you include.</p>

Related Information

[Log \[page 196\]](#)

5.3.21.1 Real-time job attributes

A real-time job has the same built-in attributes as a batch job:

Attribute	Description
Name	The name of the object. This name appears on the object in the object library and in the calls to the object.
Description	Your description of the job.
Date created	The date when the object was created.

Like batch jobs, real-time jobs use the debug and trace properties to determine what information the software collects and logs when running the job. However, real-time jobs do not support the Enable Recovery debug options.

5.3.21.2 Content of real-time job

A real-time processing loop can contain the following objects:

- A single data flow which can contain:
 - A single real-time source — XML message (required)
 - Sources — Files, XML files, and tables, including SAP tables
 - A single real-time target — XML message (required)
 - Targets — Files, XML files, tables, and template tables
 - Transforms, including queries
- Multiple data flows which can contain:
 - A single real-time source in the first data flow — XML message (required)
 - Sources — Files, XML files, and tables, including SAP tables
 - A single real-time target in the last data flow — XML message (required)
 - Targets — Files, XML files, tables, and template tables
 - Transforms, including queries
 - Memory tables — used to as staging tables to move data sets to the next data flow in the job.
- Multiple work flows, scripts, conditionals, while loops, etc.

Real-time jobs can also be built using IDocs. See the *Supplement for SAP* for more information.

i Note

Real-time jobs can not use the following features: Data_Transfer transform, run as a separate process, or run at a job distribution level lower than a job level.

5.3.21.3 Arranging metadata

If you delete an object used in a real-time job from the object library, calls to the object are replaced with an icon indicating that the calls are no longer valid.

You can insert any number of work flows and data flows into a real-time job as long as the data flow models for a real-time processing loop are followed.

- A single data flow in a loop — Must have both a message source and target
- Multiple data flows in a loop — Must have a message source in the first data flow and a message target in the last data flow

At the job level, work flows and data flows cannot be designed to run in parallel. Inside a job level work flow, they can.

The messages in a data flow are significant. The following table indicates how data flows can be used.

If a data flow has:	It can be used as the:
One XML message source	First data flow in a real-time processing loop.
One XML message target	Last data flow in a real-time processing loop.

If a data flow has:	It can be used as the:
One XML message source and one XML message target.	Only data flow in a real-time processing loop.
No message source or target	Not the first, last, or only data flow in a real-time processing loop. Data flow in a batch job.

Data flows that do not contain messages can be used in batch jobs or in real-time processing loops (between its first and last data flows).

5.3.21.4 Message processing

Unlike batch jobs, real-time jobs are designed to process multiple messages rather than just files or tables of data.

For transforms that require all of the message's data at one time, such as queries that include aggregation functions, data is cached temporarily. The transform performs the specified operation, then clears caches in preparation for the next message.

To test a real-time job using the Designer, the recommended procedure is to test one message and create a target test file to receive the data. A real-time job will clear data after processing each message if system defaults are used. Therefore, deselect:

- [Delete data from table before loading](#) for a table target
- [Delete file](#) for a flat file target
- [Delete and re-create file](#) for an XML target

5.3.21.5 Loading targets as a single transaction

In a real-time job, you can load more than one table from a single datastore in a single transaction. When transactional loading is turned on for table targets, SAP Data Services sends `INSERT` statements for any of the tables included in the transaction to the database to process. You can also control which tables are loaded first by specifying the transaction order for the tables.

If the data flow includes a real-time target, it is always loaded in parallel with other targets to ensure load time is as short as possible.


5.3.21.6 Starting and stopping real-time services

For development and testing, you can manually execute a real-time job from the Designer. The Designer runs a real-time job in test mode.

When testing a real-time service or when running in production, the Access Server triggers the Job Server to process a request using the logic you built inside a real-time processing loop. The Access Server can also

trigger the Job Server to shut down real-time processing loops. In a production environment, you control the operation of real-time services using the Administrator.

5.3.22 Script

Characteristic	Description				
	Script icon.				
Class	Single-use				
Access	With a work flow or job diagram in the workspace, click the script icon in the tool palette.				
Description	<p>A script is a single-use object that assigns values to local, global or environment variables in a job or work flow. Define the script using the SAP Data Services scripting language.</p> <p>Scripts have the following attribute:</p> <table><tr><th>Attribute</th><th>Description</th></tr><tr><td>Name</td><td>The name of the object. This name appears on the object in the diagram.</td></tr></table>	Attribute	Description	Name	The name of the object. This name appears on the object in the diagram.
Attribute	Description				
Name	The name of the object. This name appears on the object in the diagram.				

Related Information

[Scripting Language \[page 1259\]](#)

5.3.23 Source

Characteristic	Description
Class	Single-use

Characteristic	Description
Access	<ul style="list-style-type: none"> To insert a document, table, or template table as a source, open the object library, go to the Datastores tab, select the object, drag it into the workspace, and select Make Source. To insert a flat file as a source, open the object library, go to the Formats tab, select the file format template for the file, drag it into the workspace, and select Make Source. Use the file format editor to specify the file's location. To insert an XML message or file as a source, open the object library, go to the Formats tab, select the an XML Schema or DTD format template, drag it into the workspace, and select Make XML message source or Make XML file source. Use the source editor to specify the test file name for the message or the source XML name for the file. To view options of a particular source, click the name of the source in the workspace or in the project area. This opens the appropriate editor, such as the table editor, or the XML file, XML message, or flat file format editors.
Description	<p>A source is an object from which SAP Data Services reads data.</p> <p>In a batch job, a source can be a document, a file, a table, a previously defined template table, an XML file, or a source-specific data flow (see your source-specific supplement for more options).</p> <p>In a real-time job, a source can be a table, a previously defined template table, a flat file, an XML message, or an XML file. Each real-time job must have exactly one real-time data source.</p> <p>You can make an embedded data flow a source.</p> <p>Options available for sources from adapter datastores depend on the adapter implementation. Thus, options vary by data source and adapter version. See your adapter documentation for more information.</p>

5.3.23.1 Table source

You can tune performance by configuring the following common source options.

Option	Description
Make Port	Makes the source table an embedded data flow port.
Enable partitioning	Enables SAP Data Services to use the partition information in this table. If this option is selected, the software reads table data using the number of partitions in the table as the maximum number of parallel instances.

Option	Description
Join rank	<p>Indicates the rank of the source relative to other tables and files joined in a data flow. The software joins sources with higher join ranks before it joins sources with lower join ranks.</p> <p>Join rank specified in the Query transform editor FROM tab overrides any join rank specified in a source. Best practice is to specify the join rank only in the Query transform editor.</p> <p>Must be a non-negative integer. Default value is 0.</p> <p>For more information, see the Other Tuning Techniques section in the <i>Performance Optimization Guide</i>.</p>
Cache	<p>Indicates whether the software should read the required data from the source and load it into memory or pageable cache. Because an inner source of a join must be read for each row of an outer source, you might want to cache a source when it is used as an inner source in a join.</p> <p>Options are:</p> <ul style="list-style-type: none"> • Yes: The source is always cached unless it is the outer-most source in a join. • No: The source is never cached. <p>The default is Yes.</p> <p>Cache specified in the Query transform editor FROM tab overrides any cache specified in a source. Best practice is to specify the cache only in the Query transform editor.</p>
Array fetch size	<p>Indicates the number of rows retrieved in a single request to a source database. The default value is 1000. Higher numbers reduce requests, lowering network traffic, and possibly improve performance. Maximum value is 5000.</p> <p>This option is available for source tables from Amazon Redshift, DB2, Informix, ODBC, Oracle, and SQL Server datastores.</p> <p>When retrieving a column with an Oracle LONG data type, the software automatically sets Array Fetch Size to 1. If a column has an Oracle LONG data type, the software can only retrieve one row at a time.</p>

For Oracle source tables you can use an overflow file for error handling. Select **Yes** for **Use overflow file** and enter a name for the file. Errors that occur while reading data are logged into the overflow file and the job execution proceeds while ignoring the rows that cause the error. To set the location of the overflow file directory use the table's Datastore Editor.

5.3.23.2 Amazon Redshift source

Option descriptions for using an Amazon Redshift database table as a source in a data flow.

When you use an Amazon Redshift table as a source, the software supports the following features:

- All Redshift data types
- Optimized SQL
- Basic push-down functions

The following list contains behavior differences from Data Services when you use certain functions with Amazon Redshift:

- When using `add_month(datetime, int)`, pushdown doesn't occur if the second parameter is not in an integer data type.

- When using `cast(input as 'datatype')`, pushdown does not occur if you use the real data type.
- When using `to_char(input, format)`, pushdown doesn't occur if the format is 'XX' or a number such as '099', '999', '99D99', '99G99'.
- When using `to_date(date, format)`, pushdown doesn't occur if the format includes a time part, such as 'YYYY-MM-DD HH:MI:SS'.

For more information, see SAP Note [2212730](#) and “Maximizing Push-Down Operations” in the *Performance Optimization Guide*.

The following table lists source options when you use an Amazon Redshift table as a source:

Option	Description
<i>Table name</i>	Name of the table that you added as a source to the data flow.
<i>Table owner</i>	Owner that you entered when you created the Redshift table.
<i>Datastore name</i>	Name of the Redshift datastore.
<i>Database type</i>	Database type that you chose when you created the datastore. You cannot change this option.

The Redshift source table also uses common table source options.

Related Information

- [Table source \[page 208\]](#)
- [Amazon Redshift data types \[page 319\]](#)
- [Amazon Redshift target table options \[page 226\]](#)
- [Amazon Redshift \[page 70\]](#)
- [Viewing Optimized SQL \[page 969\]](#)

5.3.23.3 CDC table source

If a table comes from a CDC datastore, click the *CDC Options* tab and complete the following to set the changed-data capture options. Options vary by source database type.

Option	Description
CDC subscription name	<p>The name used to mark sets of changed data read from the continuously growing CDC table. The subscription name marks the last row read so that the next job starts reading the CDC table from that position.</p> <p>You can use multiple subscription names to identify different users who read from the same imported CDC table. The subscription saves the position of each user.</p> <p>Select from the list or type a new name to create a new subscription. A subscription name must be unique within a datastore, owner, and table name. For example, you can use the same subscription name without conflict with different tables that have the same name in the same datastore if they have different owner names.</p> <p>This value is required for the Microsoft SQL server Change Tracking and Replication Server methods.</p>
Enable check-point	<p>Enables the software to restrict CDC reads using check-points. Once a check-point is placed, the next time the CDC job runs, it reads only the rows inserted into the CDC table since the last check-point.</p> <p>By default, check-points are not enabled.</p>
Automatically delete rows after reading	<p>For Microsoft SQL Server only.</p> <p>If the option is cleared, no rows are deleted from the CDC table.</p> <p>If the option is selected, the behavior is different depending on the CDC method used:</p> <ul style="list-style-type: none"> Replication Server method: If more than one data flow uses the same CDC table as a source, only the rows that have been read by all readers are deleted. If any of those readers have the option cleared, no rows are deleted. Changed-data capture (CDC) method: All rows of the CDC table are deleted regardless of whether other readers have finished reading from the table. Therefore, if more than one data flow uses the same CDC table as a source, only select this option for the data flow that executes last.
Get before-image for each update row	<p>Some databases allow two images to be associated with an UPDATE row: a before-image and an after-image. If your source can use before-images and you want to read them during change-data capture jobs, enable this option.</p> <p>By default, only after-images are retrieved.</p>
Enable Auto Correct	<p>For a Replication Server CDC reader, select to correct records in the target for transactions that were captured by the Replication Server during the initial load. This option is useful only when the loader can identify records in the target using keys from the source table.</p> <p>The Replication Server CDC reader does the following for auto-correct processing:</p> <ul style="list-style-type: none"> For all inserted records, creates a DELETE record followed by an INSERT record. Creates a DELETE record for the before image, followed by DELETE and INSERT records for the after image for updated records that had a key value change.

5.3.23.4 HP Vertica table source

Options and descriptions for setting up an HP Vertica table as a source in a data flow.

Option	Description
<i>Table name</i>	The name of the table that you added as a source to the dataflow.
<i>Table owner</i>	The owner that you entered when you created the HP Vertica table.
<i>Datastore name</i>	The name of the HP Vertica datastore.
<i>Database type</i>	Set to HP Vertica by default. The database type that you chose when you created the datastore. You cannot change this option.

Related Information

[Database datastores \[page 68\]](#)

5.3.23.5 Flat file source

A flat file source has the *Join rank* and *Cache* options in addition to the other General file format options listed for a source table. For these two options, SAP Data Services uses the same interpretation for both files and tables.

The flat file source options are the same in the *Source File Editor* (open a source in a data flow) as in the *File Format Editor* (create new file format) except for the following file format types: Unstructured text and Unstructured binary file. there is an additional option named *Number of files to read*. This option indicates the maximum number of files to read. A value of zero reads all files. The default is blank.

You can associate an existing file location object to a flat file format. A file location object contains specific file transfer protocol information and local and remote server information, for file transfer between remote and local servers.

Related Information

[File location object \[page 171\]](#)

[Target files \[page 224\]](#)

[File format properties and modes \[page 147\]](#)

5.3.23.6 JSON file source

Source tab option descriptions for a JSON file.

A JSON file source has the same *Join rank* and *Make port* options as tables.

Source tab

Option	Description
<i>File Location</i>	<p>(Optional) Select the name of an existing file location object.</p> <p>A file location object contains file transfer protocol information (such as FTP or SFTP) and local and remote server information. The software uses this information to transfer a copy of the JSON file from the remote server to the local server to use as a source in a data flow.</p>
<i>Delete file after transfer</i>	<p>Available when you select a file location object.</p> <ul style="list-style-type: none">• Check to delete the local file copy after the software loads it as a source in the data flow.• Uncheck to save the local file copy after the software loads it as a source in the data flow.
<i>File</i>	<p>The location relative to the Job Server of a JSON-formatted file to use as the source. You can enter a variable for this option.</p> <div><p>i Note</p><p>If your Job Server is on a different computer than the Designer, you cannot choose <code><Select file></code> to Browse for the file path. You must type the path. You can type an absolute path or a relative path, but the Job Server must be able to access it.</p></div>
<i>Enable validation</i>	<p>A check box to turn on the comparison of the incoming data to the stored JSON Schema. When this option is enabled, the data flow throws an exception if the incoming source is not valid.</p>
<i>Format name</i>	<p>The name of the JSON Schema format file used in the data flow as a source.</p> <div><p>i Note</p><p>If you have included a File Location object name above, this JSON schema format file contains the file location information.</p></div>

Option	Description
<i>Include file name column</i>	<p>Determines whether to include a column in the source output that contains the source JSON file name. The default is No. Change the value to Yes when you want to identify the source JSON file in the following situations:</p> <ul style="list-style-type: none"> You specified a wildcard character to read multiple source JSON files at one time. You load from different source JSON files on different runs. <div> <p>i Note</p> <p>When you select this option, the following options appear:</p> <ul style="list-style-type: none"> File name column Modify button Column size Include path </div>
<i>File name column</i>	If the file name is included, the name of the column that holds the source file name. The default is DI_FILENAME.
<i>Modify</i>	If the file name is included, opens Column Properties dialog that enables you to modify File name column and Column size.
<i>Column size</i>	If the file name is included, the size (in characters) of the column that holds the source file name. The default is 100. If the size of the file name column is not large enough to store the file name, truncation occurs from the left.
<i>Include path</i>	If the file name is included, determines whether to include the full path name of the source file. The default is No.

Related Information

[Table source \[page 208\]](#)

[JSON Properties \[page 194\]](#)

[File location object \[page 171\]](#)

5.3.23.7 JSON message source

A JSON message source has the same [Make port](#) option as tables. The JSON message source has these options in addition to its read-only schema information:

Option	Description
Enable validation	<p>A check box to turn on the comparison of the incoming message to the stored schema. When this option is selected, the real-time job throws an exception if the incoming message is not valid.</p> <p>When you are developing a real-time job, this validation helps you to make sure sample data is both valid and well-formed. If you select this option in production, make sure to include appropriate error handling either in the SAP Data Services job or the client application to process an error caused if a data flow in the real time job receives data that does not validate against the imported format.</p>
Test file	<p>The location relative to the Job Server of a JSON-formatted file to use as the message source when you execute the job in test mode.</p> <div><p>Note</p><p>If your Job Server is on a different computer than the Designer, you cannot use Browse to specify the file path. You must type the path. You can type an absolute path or a relative path, but the Job Server must be able to access it. A variable can also be used.</p></div>
Format name	The name of the JSON Schema format used in the Designer.
Include file name column	<p>Determines whether to include a column in the source output that contains the source JSON file name. The default is No.</p> <p>Change the value to Yes when you want to identify the source JSON file in situations such as the following:</p> <ul style="list-style-type: none">You specified a wildcard character to read multiple source JSON files at one time.You load from different source JSON files on different runs.
<implied> : Join rank	The JSON message source is always the outer table in a join. You cannot change its join rank. This option is implied and does not appear in the editor.

Related Information

[Table source \[page 208\]](#)

5.3.23.8 Persistent cache source

A persistent cache source has the following options:

Option	Description
Make Port	Makes the persistent cache source an embedded data flow port.

Option	Description
Join rank	For this option, SAP Data Services uses the same interpretation for both persistent cache tables and database tables.
Cache	For this option, the software uses the same interpretation for both persistent cache tables and database tables.
Table name	The Table name box displays the name that you entered when you created the persistent cache table. You cannot change this name.
Table owner	The Table owner box displays the owner that you entered when you created the persistent cache table. You cannot change this name.
Datastore name	The Datastore name box displays the name that you entered when you created the persistent cache. You cannot change this name.
Database type	The Database type box displays the Persistent Cache option, which you cannot change.

Related Information

[Table source \[page 208\]](#)

5.3.23.9 SAP sources

SAP sources include Open Hub tables and SAP ODP sources.

For details on SAP sources, refer to the *SAP Data Services Supplement for SAP*.

5.3.23.10 SAP Vora table source options

Use an SAP Vora datastore as a source in your data flows.

The following table contains options that are specific for using an SAP Vora datastore as a source in a data flow. When you drag the datastore onto the Data Services workspace as a source, the software auto completes these options. You cannot edit these options.

For descriptions of the common options, see “Table Source.”

Source options

Option	Description
Table name	Name of the SAP Vora table.
Table owner	Name of the SAP Vora table owner.
Datastore name	Name of the SAP Vora datastore.

Option	Description
Database type	Vora.

Related Information

[SAP Vora datastore \[page 121\]](#)

[Table source \[page 208\]](#)

5.3.23.11 Teradata source

The [Teradata options](#) tab for a Teradata source includes the following modes.

- Parallel transporter API
- Parallel Transporter Export Operator
- None

The availability of [Advanced](#) options differs between modes. If an option is empty in the [Advanced](#) section, Data Services uses the default value specified at the database level.

For details on the following options, refer to your Teradata documentation.

Option	Description
General	
Clean up bulk reader directory after export	<p>Select Yes to delete all files in the bulk reader directory after successfully exporting.</p> <p>Select No to leave the files in the directory.</p>
Minimum number of sessions	Specifies the minimum number of sessions required for the Export driver job to continue. Default is one session.
Maximum number of sessions	<p>Specifies the maximum number of connections to Teradata. Must be greater than zero. Defaults to one session per available AMP.</p> <p>Use this parameter in conjunction with Number of export operator instances and Parallel process threads for performance tuning when reading from a Teradata source. For large volumes of data, more sessions allows more data to be read in parallel. Ideally this number should equal the number of AMPs.</p>
Number of export operator instances	<p>Specifies the number of instances for export operators.</p> <p>Use this parameter in conjunction with Maximum number of sessions and Parallel process threads for performance tuning when reading from a Teradata source. Multiple export instances can improve performance. Ideally this value should equal the number of CPUs.</p>

Option	Description
Tenacity hours	Specifies the number of hours the Export driver attempts to log on when the maximum number of load and export operations are already running on the Teradata database. Default is 4 hours.
Tenacity sleep	Specifies the number of minutes the Export driver pauses before attempting to log on when the maximum number of load and export operations are already running on the Teradata database. Default is 6 minutes.
<i>Data handling</i>	
Block size	Specifies the block size (in bytes) when returning data to the client.
Data encryption	<p>Select Yes to enable full security encryption of SQL requests, responses, and data.</p> <p>Select No to disable encryption.</p>
Query band session	Specifies a user-defined query band expression to be set for every SQL session.
<i>Notification</i>	
Level	<p>Indicates the level at which certain events are reported:</p> <ul style="list-style-type: none"> • Off: No notifications. Default. • Low: Notifications occur for Initialize, CLIV2/DBS Error, Exit. • Medium: Notifications occur for all events except File or OUTMODE Open and Statement Fetch Begin and End. • High: Notifications occur for all events.
Method	<p>Specifies the method for reporting events:</p> <ul style="list-style-type: none"> • None: No event logging. Default. • Message: Send events to a log (for example the EventLog on Windows). • Exit: Send the events to a user-defined notify exit routine and to the system log.
User-defined string	Specifies a user-defined string that precedes all messages sent to the system log.
User-defined exit routine	Specifies the name of a user-defined notify exit routine.
<i>Trace</i>	
Level	<p>Specifies the type(s) of diagnostic messages each instance of the driver writes to a log file. API mode writes to external log files and Export Operator mode writes to public or private logs.</p> <ul style="list-style-type: none"> • CLI: Activates the tracing function for CLIV2-related activities. • PX: Activates the tracing function for activities involving the common library. • Oper: Activates the tracing function for driver-specific activities. • Notify: Activates the tracing function for activities related to the Notify feature.
Tracing file	For API mode, specifies the name of the external log file used for trace messages.
<i>Miscellaneous</i>	

Option	Description
Parallel process threads	<p>Specifies the number of threads for parallel processing, which can improve performance by maximizing CPU usage on the Job Server computer.</p> <p>Use this parameter in conjunction with <i>Maximum number of sessions</i> and <i>Number of export operator instances</i> performance tuning when reading from a Teradata source. The data loads into buffers in Data Services, and the parallel process threads break these buffers into rows and columns. Ideally this number should equal to the number of CPUs.</p>
Logon mechanism	<p>Specifies which logon mechanism to use:</p> <ul style="list-style-type: none"> • Kerberos 5 • NT Lan Manager • Lightweight Directory Access Protocol • Simple and Protected GSSAPI Negotiation Mechanism
Logon mechanism data	Specifies additional optional logon mechanism data.
AccountId	An optional attribute that specifies the account associated with the user name (the user specified in the datastore).
Private log name	Specify the name of a log that is maintained by the Teradata Parallel Transporter Logger inside the public log. The private log contains all of the output provided by the Export operator.

Related Information

[Teradata \[page 125\]](#)

[Teradata target table options \[page 252\]](#)

5.3.23.12 XML file source

An XML file source has the same *Join rank* and *Make port* options as tables.

The options for setting up an XML file as a source are the same as the options in the XML Properties dialog.

Related Information

[XML Properties \[page 282\]](#)

5.3.23.13 XML message source

An XML message source has the same *Make port* option as tables.

The XML message source has these options in addition to its read-only XML Schema or DTD format information:

Option	Description
XML test file	<p>The location relative to the Job Server of an XML-formatted file to use as the message source when you execute the job in test mode.</p> <div>Note<p>If your Job Server is on a different computer than the Designer, you cannot use Browse to specify the file path. You must type the path. You can type an absolute path or a relative path, but the Job Server must be able to access it. A variable can also be used.</p></div>
Enable validation	<p>A check box to turn on the comparison of the incoming message to the stored XML Schema or DTD format. When this option is selected, the real-time job throws an exception if the incoming message is not valid.</p> <p>When you are developing a real-time job, this validation helps you to make sure sample data is both valid and well-formed. If you select this option in production, make sure to include appropriate error handling either in the SAP Data Services job or the client application to process an error caused if a data flow in the real-time job receives data that does not validate against the imported format.</p>
<code><implied></code> : Join rank	<p>The XML message source is always the outer table in a join. You cannot change its join rank. This option is implied and does not appear in the editor.</p>

Related Information

[Table source \[page 208\]](#)


5.3.23.14 Netezza table source

Options and descriptions for setting up a Netezza table as a source in a data flow.

Option	Description
Table name	The name of the table that you added as a source to the dataflow.
Table owner	The owner that you entered when you created the HP Vertica table.
Datastore name	The name of the HP Vertica datastore.

Option	Description
Database type	Set to HP Vertica by default. The database type that you chose when you created the datastore. You cannot change this option.
Table schema name	The name of the schema for the table that you added as a source to the dataflow.

5.3.24 Table

Characteristic	Description
	Table icon
Class	Reusable
Access	In the object library, click the Datastores tab. Expand a datastore to find the tables node. Expand this node to view the list of imported tables. Right-click and select Properties to view and edit table properties.
Description	<p>You can use a table as a source or target in a data flow.</p> <p>The Indexes tab on the Properties window for a table shows information about the table's indices. Under Index, the window lists the primary index followed by any secondary index. Select an index and the window lists the columns in that index under Column.</p> <p>The Partition tab on the Properties window displays how table metadata is partitioned. Partitions can be imported with a table or you can create metadata for them within SAP Data Services.</p> <p>The Attributes tab on the Properties window displays built-in table attributes.</p>

Related Information

[Template table \[page 274\]](#)

5.3.24.1 Table attributes

Batch jobs have the following built-in attributes:

Table Attribute	Description
Name	The name of the object. This name appears on the object in the object library and in the calls to the object.
Description	A configurable description field.
Table_Usage	A configurable label field. Use it to mark a table as <code>fact</code> or <code>dimension</code> for example.
Total_Number_Of_Rows_Proc- essed	The number of rows loaded into the table in the last successful load.
Date_last_loaded	The time the table was last successfully loaded.
Number_Of_Rows_Rejected	The number of rows rejected in the last successful load.
Number_Of_Inserts	The number of rows inserted in the last successful load.
Number_Of_Updates	The number of rows updated in the last successful load.
Date_Created	The date that the object was created.
Estimated_Row_Count	A configurable estimate of the table size used in calculating the order in which tables are read to perform join operations; used for SAP tables only.
Number_Of_Deletes	The number of rows deleted in the last successful load.
Elapsed_Time_For_Load	The time it took to load this table in the last successful load.
Table_Type	The type of datastore object for tables and hierarchies. Most often the value <code>TABLE</code> is displayed. However, the software might display the following values for SAP sources: BW master data transfer, BW transaction data transfer, BW hierarchy data transfer, SAP hierarchy.
SAP_Table_Class_Name	Imported with SAP table metadata.
Loader_Is_Template_Table	If <code>YES</code> , indicates that the table is an internal, template table created in the software. Before running production jobs, execute the job to load the target table then right-click the template table in the object library or in a data flow and the software creates the table in your database and imports it.
SavedAfterCheckOut	If <code>YES</code> , indicates you saved the table after it was checked out of the central repository. The software uses this information to determine whether to save the table in the central repository when it is checked in.
PartitionModified	If <code>YES</code> , indicates that you modified the partitions in this table using the software after you imported the table's metadata.

5.3.24.2 Column attributes for tables

SAP Data Services also supports column attributes for tables.

5.3.24.3 Viewing column attributes for a table

1. In the [Datastores](#) tab of the object library, double-click a table.

The Table Metadata window opens.

2. Right-click a column name and select [Properties](#).

The Column Properties window opens with a [General](#), [Attributes](#), and a [Class Attributes](#) tab.

3. Click the [Attributes](#) tab.

Column Attribute	Description
Business Name	A configurable logical name.
Business_ Description	A configurable business-level description of the column.
Associated_ Dimension	Set this value only if Column_Usage is set to <code>Detail</code> . The value must be in the format: <code>table.column</code> . The Detail column is created under the Dimension column you specify.
Acta_autojoin	Generated by SAP Data Services. Not configurable.
Associated_ domain	Use for databases that use domains such as PeopleSoft.
Physical_Name	Use for applications that allow logical names for a column such as Oracle Applications.

The attributes listed above are available for all tables. There may be additional attributes listed for other table types.

Related Information

[Attributes for DTDs \[page 132\]](#)

[Attributes supported for XML schemas \[page 288\]](#)











5.3.25 Target

Characteristic	Description
Class	Single-use
Access	<ul style="list-style-type: none">• To display target options, click the name of the target in the workspace or in the project area. This opens the object editor.• To display target properties, right-click a target and choose Properties.
Description	A target is an object to which SAP Data Services loads extracted and transformed data in a data flow.

5.3.25.1 Target objects

Table of objects that can be used as targets in a dataflow.

The following table contains objects that can be placed in a dataflow as a target.

Icon for target object	Target object description
	Document
	Flat file
	JSON file
	JSON message
	Nested Schemas template
	Outbound message
	Table
	Template table
	XML file
	XML message

You can make a target an embedded data flow port: set the *Make port* option to *Yes* for flat files; select the *Make port* check box for other targets.

Documents and outbound messages are only available from adapter datastores. Options available for these targets depend on the adapter implementation. Thus, options vary by data source and adapter version. See your adapter documentation for more information

5.3.25.2 Target files

You can use any flat file format as a target in a data flow. To add a target file, select a file format in the object library, drag the file format into the data flow workspace, and select *Make Target*.

If the schema defined in the file format does not match the schema that is input to the target, SAP Data Services provides validation errors to identify the mismatch.

Use the file format editor in target mode to edit the file format of a target file. You cannot edit all properties of a particular target file. You can change some properties of the file format.

You can also change the name of the target file object using the object's properties. Right-click the object and choose [Properties](#).

Related Information

[File format properties and modes \[page 147\]](#)

5.3.25.3 Target persistent cache tables

To create a new persistent cache target table, take one of the following actions:

- Select a template table under a persistent cache datastore in the object library, drag the table into the workspace, and type a name for the table.
- Click the template table icon in the tool palette, click the workspace, choose a persistent cache datastore, and type a name for the table.

Note

You cannot update a persistent cache table. If the data within it changes, you must recreate it and load it.

Target

Option	Description
Make port	Select this check box to make the target table an embedded data flow port.
Table name	The Table name box displays the name that you entered when you created the persistent cache table. You cannot change this name.
Table owner	The Table owner box displays the owner that you entered when you created the persistent cache table. You cannot change this name.
Datastore name	The Datastore name box displays the name that you entered when you created the persistent cache. You cannot change this name.
Database type	The Database type box displays <code>Persistent_Cache</code> which you cannot change.

Options

Option	Description
Column comparison	<p>Specifies how the input columns are mapped to persistent cache table columns. There are two options:</p> <ul style="list-style-type: none">• compare_by_position: SAP Data Services disregards the column names and maps source columns to target columns by position• compare_by_name: the software maps source columns to target columns by name
Include duplicate keys	Specifies whether or not to include duplicate keys in the persistent cache. Defaults to selected.

Keys

Option	Description
Key column	<p>Specify one or more columns to use as the key for the persistent cache. Click the arrow to view a drop-down list of column names.</p> <p>To change the order of the columns in the key, use one of the following options:</p> <ul style="list-style-type: none">• Right-click the column and select Move Up or Move Down.• Select the column and click the down or up arrow in the top right corner of the Keys tab. <p>To remove a column, use one of the following options:</p> <ul style="list-style-type: none">• Right-click the column and select Delete.• Select the column and click the delete icon in the top right corner of the Keys tab.

5.3.25.4 Target tables

Add a table to a data flow diagram as a target if SAP Data Services can write to the application or database containing the table.

To add a target table, select the table in the object library, drag the table into the workspace, and select [Make Target](#).

If the schema defined in the table does not match the schema that is input to the target, the software provides validation errors to identify the mismatch.

When loading DB2, ODBC, or Oracle tables, the software parameterizes the SQL. Parameterized SQL statements result in quicker load times. To parameterize SQL, the software must be able to generate, parse, and compile the statement. For example, the software is unable to parameterize SQL when using transactional loading or triggers.

You configure a target table by setting options in the target editor. Available options, like bulk loading, depend on the database in which the table is defined.

i Note

If a job using bulk load functionality fails, Data Services saves data files containing customer data in the Bulkload directory. This is done so you can review and analyze the data. The default bulk load location is %DS_COMMON_DIR%/log/BulkLoader. It is your responsibility to remove the files after analyzing them.

5.3.25.4.1 Amazon Redshift target table options

Descriptions of options for using an Amazon Redshift table as a target in a data flow.

The Amazon Redshift target supports the following features:

- input keys
- auto correct
- data deletion from a table before loading
- transactional loads

- load triggers, pre-load commands, and post-load commands
- bulk loading

When you use the bulk load feature, Data Services generates files and saves the files to the bulk load directory that is defined in the Amazon Redshift datastore. If there is no value set for the bulk load directory, the software saves the data files to the default bulk load location at: %DS_COMMON_DIR%/log/BulkLoader. Data Services then copies the files to Amazon S3 and executes the Redshift copy command to upload the data files to the Redshift table.

i Note

The Amazon Redshift primary key is informational only and the software does not enforce key constraints for the primary key. Be aware that using `SELECT DISTINCT` may return duplicate rows if the primary key is not unique.

i Note

The Amazon Redshift ODBC driver does not support parallelize load via ODBC into a single table. Therefore, the *Number of Loaders* option in the *Options* tab is not applicable for a regular loader.

Bulk loader tab

Option	Description
<i>Bulk load</i>	Select to use bulk loading options to write the data.
<i>Mode</i>	<p>Select the mode for loading data in the target table:</p> <ul style="list-style-type: none"> • <i>Append</i>: Adds new records to the table. <div data-bbox="582 1153 673 1189" data-label="Section-Header"> <h3>i Note</h3> </div> <p>Append mode does not apply to template tables.</p> <ul style="list-style-type: none"> • <i>Truncate</i>: Deletes all existing records in the table, and then adds new records.
<i>S3 file location</i>	Enter or select the path to the Amazon S3 configuration file. You can enter a variable for this option.
<i>Maximum rejects</i>	Enter the maximum number of acceptable errors. After the maximum is reached, the software stops Bulk loading. Set this option when you expect some errors. If you enter 0, or if you do not specify a value, the software stops the bulk loading when the first error occurs.
<i>Column delimiter</i>	Enter a single-character column delimiter.
<i>Text delimiter</i>	Enter a single-character text delimiter. If you insert a <i>Text delimiter</i> as well as a comma (,) for the <i>Column delimiter</i> , Data Services will treat the data file as a .csv file.

Option	Description
<i>Generate files only</i>	<p>Enable to generate data files that you can use for bulk loading.</p> <p>When enabled, the software loads data into data files instead of the target in the data flow. The software writes the data files into the bulk loader directory specified in the datastore definition.</p> <p>If you do not specify a bulk loader directory, the software writes the files to <code><DS_COMMON_DIR>\log\bulkloader\<tablename><PID></code>. Then you manually copy the files to the Amazon S3 remote system.</p> <p>The file name is <code><tablename><PID>_<timestamp>_<loader_number>_<number of files generated by each loader>.dat</code>, where <code><tablename></code> is the name of the target table.</p>
<i>Clean up bulk loader directory after load</i>	Enable to delete all bulk load-oriented files from the bulk load directory and the Amazon S3 remote system after the load is complete.
<i>Parameters</i>	Allows you to enter some Amazon Redshift copy command data conversion parameters, such as <code>escape</code> , <code>emptyasnull</code> , <code>blanksasnull</code> , <code>ignoreblanklines</code> , and so on. These parameters define how to insert data to a Redshift table. For more information about the parameters, see https://docs.aws.amazon.com/redshift/latest/dg/r_COPY.html#r_COPY-syntax-overview-optional-parameters .

General settings

Option	Description
<i>Number of loaders</i>	Sets the number of threads to generate multiple data files for a parallel load job. Enter a positive integer for the number of loaders (threads).

Related Information

[Common target table options \[page 256\]](#)

[Amazon Redshift source \[page 209\]](#)

[Amazon Redshift \[page 70\]](#)

[About Amazon Redshift datastores \[page 72\]](#)

5.3.25.4.2 DB2 target table options

The following table contains option and description information specific to DB2 target tables. All other option information for target tables can be found in the common target table options (see related links below).

Target tab

Option	Description
<i>Make port</i>	Specifies the table as an embedded data flow port.

Options tab

Option	Description
<i>Allow Merge or upsert</i>	<p>Specifies whether the Optimizer may use a MERGE statement to improve the performance of auto correct load functionality.</p> <p>Yes: Allows the Optimizer to consider using a MERGE statement during an auto correct load operation.</p> <p>If the Optimizer does not use a MERGE statement, it uses a stored procedure to identify, insert, and update rows.</p> <p>No: The Optimizer will not use a MERGE statement to improve auto correct load performance.</p> <p>The default is Yes.</p>
<i>Table type</i>	<p>Creates tables organized by row or column.</p> <p>Column Store: Create tables organized by column. Data types, blob, dbblob, and clob are not supported for this table type.</p> <p>Row store: Create tables organized by row.</p> <div data-bbox="544 1084 1366 1258"> <p>Note</p> <p>Column-organized tables are supported only on Linux (x86-x64, Intel and AMD processors) and AIX (POWER processors). There are additional steps required to set up DB2 to use the Column Store option. See Prerequisites for DB2 column-based support.</p> </div>

Bulk loader tab

Option	Description
<i>Bulk load</i>	<p>Indicate the bulk load method. Choose one of the following options:</p> <p>CLI load: Use the DB2 CLI load utility. The CLI load utility performs better than the load utility because it writes data directly from memory to the DB2 target table or view. You must use DB2 version 8.x or later.</p> <p>Import: Use the DB2 import utility to bulk load data. The import utility uses a SQL INSERT statement to write data from an input file into a table or view.</p> <p>Load: Use the DB2 bulk load utility. The load utility improves performance over the import utility by writing data directly into the data file.</p> <p>No: Do not bulk load data.</p>

Option	Description
<i>Generate files only</i>	<p>This option is available only when <i>Bulk load</i> is set to <i>import</i> or <i>load</i>.</p> <p>Select this check box to generate a data and control file. Rather than loading data into the target shown in the data flow, SAP Data Services generates a control file and a data file that you can later load using DB2 bulk loading. This option is useful when the DB2 server is located on a system running a different operating system than the Job Server.</p> <p>The software writes the data and control files in the bulk loader directory specified in the datastore definition. If you have not specified a bulk loader directory, the software writes the files in the <code><DS_COMMON_DIR>\log\bulkloader</code> directory.</p> <p>To load the data, you must manually copy the files to the remote system and start the bulk load execution.</p> <p>When you select this check box, only the <i>Text delimiter</i> and <i>Column delimiter</i> options are available.</p>
<i>Clean up bulk loader directory after load</i>	<p>Select this check box to delete all files in the bulk loader directory after the load completes successfully. If you have not specified a bulk loader directory in the <i>Connections</i> tab in the datastore definition, the software writes the files in the <code><DS_COMMON_DIR>\log\bulkloader</code> directory.</p> <p>When this option is selected, the software deletes the following files after each bulk load unless an error has occurred:</p> <ul style="list-style-type: none"> • Message file (<code>.log</code> file name) that DB2 creates for the import, load, or CLI load • Control file (<code>.ctl</code>) that the software generates only when <i>Bulk load</i> is set to <i>import</i> or <i>load</i> • Data file (<code>.dat</code>) that the software generates only when <i>Bulk load</i> is set to <i>import</i> or <i>load</i> • Bad file (<code>.bad</code>) that DB2 generates only when <i>Bulk load</i> is set to <i>load</i> and the <i>Data file on client machine</i> option is not checked. <p>If the <i>Data file on client machine</i> option is checked, DB2 creates the <code>.bad</code> file on the DB2 server working directory (specified in the <i>DB2 Properties</i> tab of the datastore definition). In this case, the software does not delete the <code>.bad</code> file when the bulk load completes.</p>

Option	Description
<i>Mode</i>	<p>Specify the mode for loading data in the target table. Available modes depend on the bulk load method.</p> <p>Available modes when <i>Bulk load</i> is set to <i>import</i>:</p> <p><i>Insert</i>: Adds new records to the table. Use when loading data into an empty table or when appending data to an existing table that contains data that you want to maintain.</p> <p><i>Insert-update</i>: If a record with matching primary keys exists in the table, updates that record; otherwise, adds new record to the table. This method requires that the target table has primary keys.</p> <p><i>Replace</i>: Deletes all existing records in the table, then adds new records.</p> <p><i>Truncate</i>: Deletes all existing records in the table, then adds new records.</p> <p>Available modes when <i>Bulk load</i> is set to <i>CLI load</i> or <i>load</i>:</p> <p><i>Insert</i>: Appends the new records into the target table.</p> <p><i>Replace</i>: Deletes the existing records, and then inserts the loaded data.</p>
<i>Rows per commit</i>	<p>Enter the number of rows that will be loaded before a commit takes place. If no value is entered, the load utility uses the default value at run time.</p> <p>This option is available only when <i>Bulk load</i> is set to <i>import</i>.</p>
<i>Save count</i>	<p>Enter the minimum number of rows loaded before the load utility establishes a consistency point. This is the point when DB2 saves the data. DB2 converts this value to a page count, and rounds up to intervals of the extent size. If you enter zero, the load utility establishes no consistency points.</p> <p>This option is available only when <i>Bulk load</i> is set to <i>load</i>.</p>
<i>Warning row count</i>	<p>Enter the number of warnings allowed for each load operation.</p> <p>This option is available only when <i>Bulk load</i> is set to <i>CLI load</i> or <i>load</i>.</p>
<i>Text delimiter</i>	<p>Enter a single character string delimiter. The default value is a double quote ("). The specified character is used in place of double quotation marks to enclose a character string. The specified character can be any printable or non-printable ASCII character, escaped with a double slash "\\".</p> <p>This option is available only when <i>Bulk load</i> is set to <i>import</i> or <i>load</i>.</p>
<i>Column delimiter</i>	<p>Enter a single-character column delimiter. The default value is a comma (,).</p> <p>This option is available only when <i>Bulk load</i> is set to <i>import</i> or <i>load</i>.</p>
<i>Maximum bind array</i>	<p>Enter the maximum number of rows extracted or transformed before the software sends the data to the DB2 table or view.</p> <p>If you do not enter a value, the software uses the default DB2 CLI Loader value which is 10000.</p> <p>This option is available only when <i>Bulk load</i> is set to <i>CLI load</i>.</p>

Option	Description
Exception table name	<p>Enter the table into which the DB2 server loads rows that violate a table constraint. Rows that violate constraints are deleted from the target table and inserted into the exception table.</p> <p>This option is available only when Bulk load is set to CLI load or load.</p>
Recoverable	<p>Select this check box to support data recovery through the DB2 roll-forward recovery feature.</p> <p>When this option is not selected, you cannot recover from failure using DB2 roll-forward.</p> <p>When this option is selected, DB2 writes a backup copy of the loaded data. You can use DB2 roll-forward recovery after failure. You must specify the directory for writing the backup file (Copy target directory). Select this option only if your DB2 target database is roll-forward enabled.</p> <p>This option is available only when Bulk load is set to CLI load or load.</p>
Copy target directory	<p>Enter the directory where copy files are stored when the Recoverable option is enabled. Only local media is supported.</p> <p>This option is available only when Bulk load is set to CLI load or load.</p>
Data file on client machine	<p>Select this check box to have the load utility process the local file directly rather than using FTP to send the data file to the DB2 server. To use this option:</p> <ul style="list-style-type: none"> • You must use DB2 version 7.x or later. • The target DB2 cannot be a DB2 enterprise (extended edition environment). • The target table and database must not be partitioned. <p>This option is applicable only if the software and DB2 are on different servers.</p> <p>This option is available only when Bulk load is set to load.</p>

Related Information

[Common target table options \[page 256\]](#)

[Prerequisites for DB2.10.5 column-based support \[page 232\]](#)

5.3.25.4.2.1 Prerequisites for DB2 10.5 column-based support

Column-organized tables are supported only on Linux (x86-x64, Intel and AMD processors) and AIX (POWER processors). There are two ways to enable this feature:

Use default table organization

Set the default table Organization to column automatically by setting the registry variable, [DB2_WORKLOAD](#) to [ANALYTICS](#). If this is set, and if you want to create a row based table, you have to explicitly specify ORGANIZE BY ROW in the create table statement.

Configure the database for analytics workload

If the [DB2_WORKLOAD](#) cannot be set to [ANALYTICS](#), do the following to create and optimally configure the database for analytics workload.

1. Set the `dft_table_org` (default table organization for user tables) database configuration parameter to COLUMN so that new tables are created as column-organized tables by default; otherwise, the ORGANIZE BY COLUMN clause must be specified on each CREATE TABLE statement
2. Set the `dft_degree` (default degree) database configuration parameter to ANY
3. Set the `dft_extent_sz` (default extent size) database configuration parameter to 4
4. Ensure that the database configuration parameters, [sortheap](#) (sort heap) and [sheapthres_shr](#) (sort heap threshold for shared sorts) are **not** set to [AUTOMATIC](#). Consider increasing these values significantly for analytics workloads. A reasonable starting point is to set `sheapthres_shr` to the size of the buffer pool (across all buffer pools). Set `sortheap` to some fraction (for example, 1/20) of `sheapthres_shr` to enable concurrent sort operations.
5. Set the [util_heap_sz](#) (utility heap size) database configuration parameter to **1,000,000** pages and [AUTOMATIC](#) to address the resource needs of the LOAD command. If the database server has at least 128 GB of memory, set [util_heap_sz](#) to **4,000,000** pages. If concurrent load operations are running, increase the value of [util_heap_sz](#) to accommodate higher memory requirements.
6. Set the [auto_reorg](#) (automatic reorganization) database configuration parameter to [ON](#).
7. Ensure that the `sheapthres` database manager configuration parameter is set to 0 (this is the default value).

Note

This setting applies to all databases in the instance.

8. Ensure that intraquery parallelism, which is required to access column-organized tables, is enabled. Intraquery parallelism can be enabled at the instance level, database level. For details, see Intraquery parallelism and intrapartition parallelism.
9. Enable concurrency control on the SYSDEFAULTMANAGEDSUBCLASS service subclass by issuing the following statement: **ALTER THRESHOLD SYSDEFAULTCONCURRENT ENABLE.**

Additional restrictions

Schemas that include column-organized tables cannot be transported.

For data replication with column-organized tables as either source or target, there are some sql statements that do not support column-organized tables as source and target tables:

- SET INTEGRITY
- CREATE TRIGGER
- CREATE EVENT MONITOR
- CREATE INDEX AND ALTER INDEX

5.3.25.4.3 Google BigQuery target table

Option descriptions for the *Target* tab in the datastore explorer for the Google BigQuery datastore table.

When you include a Google BigQuery table in a data flow, you edit the target information for the target table. Double-click the target table in the data flow to open the target editor.

Options specific to Google BigQuery

Option	Description
<i>Make Port</i>	<p>Creates an embedded data flow port from a source or target file.</p> <p>Default is No. Choose Yes to make a source or target file an embedded data flow port.</p> <p>For more information, see “Creating embedded data flows” in the <i>Designer Guide</i>.</p>
<i>Mode</i>	<p>Designates how Data Services updates the Google BigQuery table. The default is Truncate.</p> <ul style="list-style-type: none"> • <i>Append</i>: Adds new records generated from Data Services processing to the existing Google BigQuery table. • <i>Truncate</i>: Replaces all existing records from the Google project table with the uploaded data from Data Services.
<i>Number of loaders</i>	<p>Sets the number of threads to use for processing.</p> <p>Enter a positive integer for the number of loaders (threads).</p> <p>Each loader starts one resumable load job in Google BigQuery to load data.</p> <p>Loading with one loader is known as single loader loading. Loading when the number of loaders is greater than 1 is known as parallel loading. You can specify any number of loaders.</p>
<i>Maximum failed records per loader</i>	<p>Sets the maximum number of records that can fail per loader before Google stops loading records. The default is zero (0).</p>

The *Target* tab also displays the Google table name and the datastore used to access the table.

5.3.25.4.4 HP Vertica target table options

Options and descriptions for setting up an HP Vertica table as a target in a data flow.

General

Option	Description
<i>Column comparison</i>	<p>Default is <i>Compare by name</i>.</p> <p>Specifies how the input columns are mapped to output columns. There are two options:</p> <ul style="list-style-type: none">• <i>Compare by position</i>: The software disregards the column names and maps source columns to target columns by position.• <i>Compare by name</i>: The software maps source columns to target columns by name. <p>Validation errors occur if the data types of the columns do not match.</p>
<i>Number of loaders</i>	<p>The default number of loaders is <i>1</i>, which is single loader loading.</p> <ul style="list-style-type: none">• Single loader loading: Loading with one loader.• Parallel loading: Loading when the number of loaders is greater than one. <p>When parallel loading, each loader receives the number of rows indicated in the <i>Rows per commit</i> option. Each loader applies the rows in parallel with the other loaders.</p> <p>For example, if you choose a <i>Rows per commit</i> of 1000 and set the <i>Number of Loaders</i> to 3, the software loads data as follows:</p> <ul style="list-style-type: none">• Sends the first 1000 rows to the first loader• Sends the second 1000 rows to the second loader• Sends the third 1000 rows to the third loader• Sends the next 1000 rows back to the first loader
Error handling	
Option	Description
<i>Use overflow file</i>	<p>Default is <i>No</i>.</p> <p>This option is used for recovery purposes. If the software cannot load a row, the row is written to a file. When this option is set to <i>Yes</i>, options are enabled for the file name and file format.</p>

Update control

Option	Description
<i>Use input keys</i>	<p>Default is <i>No</i>.</p> <p><i>Yes</i>: If the target table does not contain a primary key, this option enables the software to use the primary keys from the input.</p> <p><i>No</i>: If the target is a Microsoft SQL Server database table, and the identity column is mapped as the primary key, this option must = No.</p>
<i>Update key columns</i>	<p>Default is <i>No</i>.</p> <p><i>Yes</i>: The software updates key column values when it loads data to the target.</p>
<i>Auto correct load</i>	<p>Default is <i>No</i>.</p> <p><i>Yes</i>: Use auto correct loading. Auto correct loading ensures that the same row is not duplicated in a target table. This is particularly useful for data recovery operations.</p> <div data-bbox="821 1008 914 1043" data-label="Section-Header"> <h3>i Note</h3> </div> <div data-bbox="821 1061 1383 1126" data-label="Text"> <p>This option is not available for targets in real time jobs or target tables that contain LONG columns.</p> </div> <p>When you select <i>Yes</i> for this option, the software reads a row from the source, then checks if the row exists in the target table with the same values in the primary key. If <i>Use input keys</i> is set to <i>Yes</i>, the software uses the primary key of the source table. Otherwise, the software uses the primary key of the target table. If the target table has no primary key, the software considers the primary key to be all the columns in the target.</p> <p>If a matching row does not exist, the software inserts a new row, regardless of other options.</p> <p>If a matching row exists, the software updates the row depending on the value of <i>Ignore columns with value</i>.</p> <p>When the column data from the source matches the value in <i>Ignore columns with value</i>, the software does not update the corresponding column in the target table. The value may be spaces. Otherwise, the software updates the corresponding column in the target with the source data.</p>

Option	Description
Ignore columns with value	Enter a value that might appear in a source column and that you do not want updated in the target table. The value must be a string, it can include spaces, but the string cannot be in single or double quotations. When this value appears in the source column, the software does not update the corresponding target column during auto correct loading.

Transaction control

Option	Description
Include in transaction	<p>Default is No.</p> <p>Yes: Indicates that this target is included in the transaction processed by a batch or real-time job. This option allows you to commit data to multiple tables as part of the same transaction. If loading fails for any one of the tables, the software does not commit an data to any of the tables. The tables must be from the same datastore.</p> <p>Transactional loading can require rows to be buffered to ensure the correct load order. If the data being buffered is larger than the virtual memory available, the software reports a memory error.</p> <p>If you choose to enable transactional loading, the following options are not available:</p> <ul style="list-style-type: none"> • Rows per commit • Use overflow file and overflow file specification • Number of loaders <p>The software does not push down a complete operation to the database if transactional loading is enabled.</p>

5.3.25.4.5 Informix target table options

The following table contains option and description information specific to Informix target tables. All other option information for target tables can be found in the common target table options (see related links below).

i Note

Commit at the end of INSERT..SELECT option is not applicable.

Options

Option	Description
Drop and re-create table	Drops the existing table and creates a new one with the same name before loading.

Bulk Loader Options

Option	Description
Bulk load	Select this check box to use Informix bulk loading options to write the data.
Generate files only	<p>Select this check box to generate a data and control file. Rather than loading data into the target shown in the data flow, SAP Data Services generates a control file and a data file that you can later load using Informix bulk loading. This option is often useful when the Informix server is located on a system running a different operating system than the Job Server.</p> <p>The software writes the data and control files in the bulk loader directory specified in the datastore definition. If you have not specified a bulk loader directory, the software writes the files in the <code><DS_COMMON_DIR>\log\bulkloader</code> directory.</p> <p>You need to copy the files to the remote system manually. The files names are <code><tablename>.ctl</code> and <code><tablename>.dat</code>, where <code><tablename></code> is the name of the target table.</p>
Lock table	Select this check box to lock the table for the duration of the load.
Clean up bulk loader directory after load	<p>Select this check box to delete all bulk load-oriented files after the load is complete, unless an error occurs. When you select this option, the software deletes these files after a successful bulk load:</p> <ul style="list-style-type: none"> • Control file • Log file • Bad file
Mode	<p>Select the mode for loading data in the target table:</p> <ul style="list-style-type: none"> • <i>Append</i>: Adds new records to the table. • <i>Replace</i>: Deletes all existing records in the table and then adds new records.
Bulk loader server name	Enter the name of the Informix database server.
Bulk loader database name	Enter the name of the target information warehouse database.
Rows per commit	Enter the number of rows that must be loaded before a commit takes place.
Field delimiter	Enter the character that separates columns. Make sure the character you designate is not used in any of the data columns.
Maximum rejects	Enter the maximum number of acceptable warnings. Bulk load stops after this many warnings. Set this parameter when you expect no warnings, but want to verify that the correct file and table are used. If you enter 0, or do not specify a value, the load continues regardless of the number of warnings issued. The default value is 10.
Begin/end column character	Enter the character that designates the beginning or ending of the column.

Related Information

[Common target table options \[page 256\]](#)

5.3.25.4.6 Microsoft SQL Server target table options

The following table contains option and description information specific to Microsoft SQL Server target tables.

All other option information for target tables can be found in the [Common](#) target table options. Bulk loader options are not available for Microsoft APS. Only the bulk load options [Mode](#), [File location](#), [Maximum rejects](#), and [File options](#) are available for Microsoft Azure DW.

Options tab

Option	Description
Identity Insert	<p>Appears under the General section.</p> <p>Target table options specifies whether to enable the software to insert identity column content to the Microsoft SQL Server database target table.</p> <p>On: Identity column values are generated by the software. You must include the identity column in the output map.</p> <p>Off: Identity column values are generated by SQL Server.</p> <div><p>i Note</p><p>Make sure the setting that you make for this option is compatible with other related settings to avoid validation errors:</p><ul style="list-style-type: none">• If Identity Insert is set to On, you must map the identity column in the target or else the software issues a validation error.• If the identity column is a key column, you must set the option Update Key Column in the Options tab to No. If you set it to Yes, the software issues a validation error.• The Identity Insert option must be set to Off when you are using the transaction loader or the software issues a validation error.</div> <div><p>i Note</p><p>If you set Include in transaction under Transaction Control to Yes, the software automatically sets the Identity Insert option to Not Applicable.</p></div> <div><p>i Note</p><p>The Identity Insert option is not supported for Microsoft APS or Azure DW.</p></div>

Option	Description
<i>Allow merge or upsert</i>	<p>This option is available only for Microsoft SQL Server 2008 and higher.</p> <p>Appears under the <i>Update control</i> section.</p> <p>Specifies whether the Optimizer may use a MERGE statement to improve the performance of auto correct load functionality.</p> <p>Yes: (The default setting.) Allows the Optimizer to consider using a MERGE statement during an auto correct load operation. If the Optimizer does not use a MERGE statement, it uses a T-SQL block to identify, insert, and update rows.</p> <p>No: The Optimizer will not use a MERGE statement to improve auto correct load performance.</p> <div> <p>i Note</p> <p>When the data flow contains a Query transform with an Order by clause, the Optimizer always uses a T-SQL block to identify, insert, and update rows.</p> </div> <div> <p>i Note</p> <p>The <i>Allow merge or upsert</i> option is not supported for Microsoft APS or Azure DW.</p> </div>
Bulk loader options	
Option	Description
<i>Bulk load</i>	<p>Select this check box to use Microsoft SQL Server bulk loading options to write the data.</p> <div> <p>i Note</p> <p>The <i>Bulk load</i> option is not supported for Microsoft APS.</p> </div> <div> <p>i Note</p> <p>In Azure DW, if you insert a NULL in a DW decimal column using bulk loading, an error will occur.</p> </div>
<i>Rows per commit</i>	<p>Select this option to load a specified number of rows per commit.</p>

Option	Description
<i>Mode</i>	<p>Specify the mode for loading data in the target table:</p> <ul style="list-style-type: none"> • <i>Append</i>: Adds new records to the table. • <i>Truncate</i>: Deletes all existing records in the table, and then adds new records.
<i>Maximum rejects</i>	<p>When the value for <i>Maximum rejects</i> is exceeded, the bulk loader stops. Set this parameter when you expect no warnings about rejected rows, but want to verify that the correct file and table are being loaded. If you enter 0 or do not specify a value, the load continues regardless of the number of warnings issued.</p> <p>A rejected row contains data that does not match the expected format or information specified for the table being loaded.</p> <div> <p>i Note</p> <p>In Azure DW, if the number of rejected rows is lower than the number of maximum rejects, the job will complete successfully without a warning message and also without a listing of the rows that failed during the load.</p> </div>
<i>Network packet size</i>	<p>Specify the network packet size in KB. Default is 4 KB. When loading, the client caches rows until it either fills a network packet or reaches the commit size (regardless of whether the packet is full). Then the client sends the packet to the server. You can affect performance by tuning commit size and network packet size.</p>
Options specific to Microsoft Azure DW:	
<i>File Location</i>	<p>The menu displays a list of Azure blob storage and Data Lake Store file locations that have been previously set up as global variables or system parameters. Select the file location you want to use from the drop-down menu.</p>
<i>File Options</i>	<p>Enter the delimiters you want to use for both columns and text. The default value for the Number of loaders is 4. Increasing or decreasing this number may affect the bulk load performance. The software will save the files in the bulkload directory. If you check <i>Clean up bulk loader directory after load</i>, the software will delete the data file(s) from the local bulkload directory and Azure blob storage or Data Lake Store after the job completes successfully.</p>

Related Information

[Common target table options \[page 256\]](#)

[Update Control \[page 262\]](#)

[Transaction Control \[page 265\]](#)

5.3.25.4.7 Netezza target table options

The following table contains option and description information specific to Netezza target tables. All other option information for target tables can be found in the common target table options (see related links below).

i Note

All options under the Transaction Control category are not available for Netezza (Include in Transaction, Transaction ID, Commit at the end of INSERT...SELECT).

Bulk Loader options tab

Option	Description
Bulk load	Select a bulk-loading method: <i>File</i> : Writes all data to a file before loading through the external table to the staging table. For files that are smaller than 4 GB in size, select this option for faster performance. <i>Named pipe</i> : Streams data as it is written to the named pipe through the external table to the staging table. For files that are larger than 4 GB in size, select this option for faster performance. <i>None</i> : Does not use bulk loading (default).
Mode	Specify the mode for loading data to the target table: <i>Append</i> : Adds new records to table (default). <i>Truncate</i> : Deletes all existing records in table then adds new records.
Update method	Specify how to apply UPDATE SQL. <i>Delete-insert</i> : Issues a DELETE to the target table for data that matches the old data in the staging table, and then issues an INSERT with the new data (default). Select this option for faster performance. i Note Do not use this option if the update rows do not contain data for all columns in the target table because SAP Data Services will replace missing data with NULLs. <i>Update</i> : Issues an UPDATE to the target table.
Maximum rejects	Enter the maximum number of error records allowed before the software terminates the job. Default is 10.

Option	Description
Commit size	Enter the maximum number of rows loaded to the staging and target tables before saving the data. A value of 0 means the software loads and saves all rows automatically (default). This option overrides the Rows per commit setting.
Text delimiter	Enter the character used to delimit (enclose) char or varchar data. Valid values are single quotes, double quotes, or blank (default). If you expect the data to contain single or double quotes, use an Escape character.
Field delimiter	Enter the character that separates the columns in a row. Valid values are all printable characters except the single quote. The default is pipe (). If you expect the field delimiter character to be present in the varchar or char data, use a Text delimiter.
Generate files only	Select this check box to generate data into a file on the Job Server. This option is useful when you want to manually load data using the SQL statements generated in the .sql script. Default is cleared.
Escape character	<p>If you expect the Text delimiter character or a row delimiter (the ASCII character line feed, LF) to be present in any varchar or char data, enter a backslash (the only valid value) to escape the text or row delimiter before writing to the file or named pipe. (The software also escapes back slashes in the data with a back slash.)</p> <p>If you expect a NULL string as part of varchar data, then you should set the escape character. The NULL string is used as null indicator to identify null values while bulk loading.</p> <div> <p>Note</p> <p>The null indicator string is not case sensitive so if you have varchar data as NULL (case insensitive) the escape character should be set.</p> </div> <p>Default is cleared because this option can degrade job execution performance significantly.</p>
Clean up bulk loader directory after load	<p>Select this check box to delete all bulk-loader-related files (for example dat, sql, nzlog, nzbad, etc.) after the load completes successfully.</p> <p>If an error occurs during the bulk load, the Netezza server creates an nzbad file in the Database server working directory defined in the datastore editor.</p> <p>If you have enabled FTP by configuring the FTP options in the datastore editor, the software transfers the nzbad and nzlog files from the Database server working directory to the Bulk loader directory on the Job Server computer. Default is checked.</p>

Related Information

[Common target table options \[page 256\]](#)

5.3.25.4.8 ODBC target table options

The following table contains option and description information specific to ODBC target tables. All other option information for target tables can be found in the common target table options (see related links below).

i Note

Bulk loader options are only visible if you are loading to a Netezza target.

Option	Description
Rows per commit	Enter the number of rows that will be loaded before a commit takes place. If no value is entered, the load utility uses the default value at run time. This option is available only when <i>Bulk load</i> is set to <i>import</i> .

Related Information

[Common target table options \[page 256\]](#)

[Netezza target table options \[page 242\]](#)

5.3.25.4.9 Oracle target table options

The following table contains option and description information specific to Oracle target tables. All other option information for target tables can be found in the common target table options (see related links below).

Options

Option	Description
<i>Allow merge</i>	Specifies whether the Optimizer may use a MERGE statement to improve the performance of auto correct load functionality. <i>Yes</i> : Allows the Optimizer to consider using a MERGE statement during an auto correct load operation. If the Optimizer does not use a MERGE statement, it uses a PL/SQL block to identify, insert, and update rows. <i>No</i> : The Optimizer will not use a MERGE statement to improve auto correct load performance. The default is Yes.

Bulk loader options tab

Option	Description
<i>Bulk load</i>	<p>Select a bulk loading method. Use database tools to load data in bulk instead of using SQL statements.</p> <p><i>API</i>: Allows you to use an Oracle direct-path load API to load table data directly to database files. The target database must be Oracle 8.1 or later.</p> <p><i>File</i>: Allows you to use a staging file and the Oracle SQL*Loader to load table data. To use this feature, the version of the Oracle SQL*Loader (specified on this tab) and the database (specified in the datastore for the target) must match. If you also want to use the direct-path load method, manually select it from the File Options section. Otherwise, a conventional load is performed. Conventional loads generally are slower than direct-path loads because data is loaded to tables rather than directly to database files associated with tables.</p> <p><i>None</i>: Allows you to use normal load functionality. See the Options tab.</p>
<i>Mode</i>	<p>Specify the mode for loading data in the target table.</p> <p>Available modes when Bulk load is set to File:</p> <p><i>Append</i>: Adds new records containing the loaded data.</p> <p><i>Insert</i>: Adds new records containing the loaded data. Requires that the table is empty before loading. SQL loader terminates with an error if the table is not empty.</p> <p><i>Replace</i>: Deletes all existing records in the table, and then adds new records containing the loaded data.</p> <p><i>Truncate</i>: Deletes all existing records in the table, and then adds new records containing the loaded data. This mode does not execute any delete triggers.</p> <p>Available modes when Bulk load is set to API:</p> <p><i>Append</i>: Adds new records containing the loaded data.</p> <p><i>Truncate</i>: Deletes all existing records in the table, and then adds new records containing the loaded data.</p>
<i>Rows per commit</i>	<p>Specifies the transaction size in number of rows or bulk loading. If Rows per commit is set to 1000, a commit is sent to the underlying database every 1000 rows. If you do not enter a value, the default (1000) is used.</p> <p>Bulk loading is not available for targets in a real-time jobs.</p>
<i>Maximum rejects</i>	<p>Enter the maximum number of error records allowed before the job is terminated.</p> <p>If you do not enter a value, the default (10) is used.</p>
<i>Recoverable</i>	<p>Select this check box to log direct-path information about the loaded data in the Oracle redo log.</p> <div> <p>i Note</p> <p>All conventional loads are automatically specified as recoverable.</p> </div>
<i>Use local index to rebuild primary key</i>	<p>(API method only) Select this option to ensure that the primary key uses the partitioned index (if the table has local partitioned indexes on primary key) instead of any other indexes that are available on primary key.</p>

Option	Description
<i>SQL *Loader version</i>	(File method only) The version used to load data into the table. The version of the Oracle SQL*Loader and the database (specified in the datastore for the target) must match.
<i>Text delimiter</i>	(File method only) Enter the character used to delimit char or varchar columns. The default character is a double quotation mark ("). Make sure the character you enter is not used in any of the data columns.
<i>Field delimiter</i>	<p>(File method only) Enter the character used to separate columns. The default character is a comma (.). Make sure the character you enter is not used in any of the data columns.</p> <p>You can specify a non-printable character by entering the ASCII equivalent, such as:</p> <p>/ASCII_number</p>
<i>Maximum bind array</i>	<p>(File method only) Enter the maximum bind array. The bind array needs to be large enough to contain a single row. For good performance, make this large enough to hold 100 rows.</p> <p>If you do not enter a value, the default Oracle Bulk Loader value is used.</p>
<i>Use the control file</i>	<p>(File method only) Select this check box to load data from a specific bulk loading control file and data file. Rather than loading data from the source shown in the data flow, Data Services directs Oracle to load data from the data file associated with the named control file.</p> <p>Enter the name of the control file. Do not include the .ctl extension in the file name.</p> <p>If you do not specify a complete path, Data Services searches for the file in:</p> <ul style="list-style-type: none"> • The path you have specified as the bulk loader directory in the datastore definition • \\${LINK_DIR}\log\bulkloader <p>Both a control file and an associated data file must be in the same directory.</p> <p>If you select this option, you can also select these options or specify these values:</p> <ul style="list-style-type: none"> • Direct path • Clean up bulk loader directory after load • Rows per commit • Maximum rejects • Maximum bind array <p>A variable can also be used.</p>
<i>Generate files only</i>	<p>(File method only) Select this check box to have Data Services generate a data and control file. Rather than loading data into the target shown in the data flow, Data Services generates a control file and a data file that you can later load using Oracle bulk loading. This option is often useful when the target database is located on a system running a different operating system than the Data Services Job Server.</p> <p>Data Services writes the data and control files in the bulk loader directory specified in the datastore definition. You need to copy the files to the remote system manually. Data Services names these files <tablename> .ctl and <tablename> .dat, where <tablename> is the name of the target table.</p>

Option	Description
Direct path	<p>(File method only) Select this check box to specify a direct-path load.</p> <p>To use direct-path load, the version of SQL*Loader available to the Job Server executing the job must be the same as the target database version. For example, you cannot perform a SQL*Loader Version 7.1.2 direct path load to load into a Oracle Version 7.1.3 database. For more information, see the Oracle server documentation.</p>
Clean up bulk loader directory after load	<p>(File method only) Select this check box to have Data Services delete all bulk loader-related files (control file, datafile, log file) after the load is complete. If an error occurs during the bulk load, Data Services creates a <code>.bad</code> file and does not delete any files. Errors occur when:</p> <ul style="list-style-type: none"> Log file was not created Log file contains "ORA-" or "SQL*Loader-"
Trailing nullcols	<p>(File method only) Select this check box to indicate that columns not represented in the data being loaded should be treated as null columns. Use when a data record is not complete but the existing data needs to be loaded. If this option is not selected, the system generates an error.</p>

Related Information

[Common target table options \[page 256\]](#)

5.3.25.4.10 SAP HANA target table options

Use SAP HANA tables as targets in a data flow when applicable, and complete the options specific to SAP HANA.

Options

Option	Description
Table type	<p>For template tables, select the appropriate table type for your SAP HANA target:</p> <ul style="list-style-type: none"> Column Store (default) Row Store

Bulk loading

Option	Description
Bulk load	Select to enable bulk loading.
Mode	<p>Specify the mode for loading data to the target table:</p> <ul style="list-style-type: none"> Append: Adds new records to the table (default). Truncate: Deletes all existing records in the table and then adds new records.

Option	Description
<i>Commit size</i>	<p>default: Data Services identifies the SAP HANA target table type and applies a default commit size for the maximum number of rows loaded to the staging and target tables before saving the data (committing):</p> <ul style="list-style-type: none"> • <i>Column Store</i>: commit size is 10,000 • <i>Row Store</i>: commit size is 1,000 <p>You can also type any value in the field that is greater than one.</p>
<i>Update method</i>	<p>Specify how the input rows are applied to the target table:</p> <p>Default: Data Services applies the default value for this option based on the SAP HANA target table type:</p> <ul style="list-style-type: none"> • <i>Column Store</i> tables use UPDATE. • <i>Row Store</i> tables use DELETE-INSERT. • <i>UPDATE</i>: Issues an UPDATE to the target table. • <i>DELETE-INSERT</i>: Issues a DELETE to the target table for data that matches the old data in the staging table, and then issues an INSERT with the new data. <div style="background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p>Note</p> <p>Do not use DELETE-INSERT if the update rows do not contain data for all columns in the target table, because Data Services will replace missing data with NULLs.</p> </div>

Related Information

[Common target table options \[page 256\]](#)

[Reference Guide: description of objects, target, Target tables \[page 226\]](#)

5.3.25.4.11 SAP ASE target table options

The following table contains option and description information specific to SAP ASE target tables. All other option information for target tables can be found in the common target table options (see related links below).

Bulk loader options

Option	Description
Mode	<p>Specify the mode for loading data in the target table:</p> <p><i>Append</i>: Adds new records to the table.</p> <p><i>Truncate</i>: Deletes all existing records in the table and then adds new records.</p>
Bulk load	Select this check box to use SAP ASE bulk loading options to write the data.

Option	Description
Rows per commit	Select this option to load a specified number of rows per commit.
Maximum rejects	<p>When the value for Maximum rejects is exceeded, the bulk loader stops. Set this parameter when you expect no warnings about rejected rows, but want to verify that the correct file and table are being loaded. If you enter 0 or do not specify a value, the load continues regardless of the number of warnings issued.</p> <p>A rejected row contains data that does not match the expected format or information specified for the table being loaded.</p>
Network packet size	Specify the network packet size in KB. Default is 4 KB. When loading, the client caches rows until it either fills a network packet or reaches the commit size (regardless of whether the packet is full). Then the client sends the packet to the server. You can affect performance by tuning commit size and network packet size.

Related Information

[Common target table options \[page 256\]](#)

5.3.25.4.12 SAP Sybase IQ target table options

The following table contains option and description information specific to SAP Sybase IQ target tables. All other option information for target tables can be found in the common target table options (see related links below).

Bulk loader options

Option	Description
Bulk load	Select this check box to use SAP Sybase IQ bulk loading options to write the data.
SAP Sybase IQ checkpoint	<p>If selected, Data Services enables the SAP Sybase IQ checkpoint as part of the LOAD TABLE SQL statement used to execute the bulk load.</p> <p>If cleared (the default), the checkpoint is not enabled.</p>
Binary format	<p>Select the check box to load the staging and target tables in binary format. This format generally provides faster performance.</p> <p>Clear the check box to load the staging and target tables in delimited format. This format is useful if you want to be able to view the data.</p>
Ignore conversion error	<p>Select to set the SAP Sybase IQ database option <code>CONVERSION_ERROR</code> to OFF.</p> <p>If cleared, Data Services executes with the current database value for the option. For more information, see the SAP Sybase IQ documentation.</p> <p>Conversion errors are reported as warnings in Data Services.</p>

Option	Description
Mode	<p>Specify the mode for loading data into the target table:</p> <p><i>Append</i>: Adds new records to the table.</p> <p><i>Truncate</i>: Deletes all existing records in the table and then adds new records.</p>
Block size (bytes)	The number of bytes read per block. Inappropriate adjustments to this option can dramatically affect performance. Defaults to 500,000 bytes.
File Options	
Field delimiter	This is the delimiter that separates the columns in a row. It can only be a single character (including a non-printable character) and can be represented by a string of ASCII numbers preceded by a forward slash (for instance /95 results in an underscore character while /09 represents a tab). Default value is /127.
Row delimiter	A character sequence that indicates where one row of data ends and the next begins. It can only be a single character; however, you can represent it with a string of up to 4 hexadecimal ASCII characters. Defaults to \n. Field and row delimiters cannot be the same value.
Generate files only	<p>If checked, the software generates the data files in the bulk loader directory specified in the datastore editor and does not execute the bulk load.</p> <p>If cleared (the default), the software generates the data files and executes the bulk load.</p>
Clean up bulk loader directory after load	<p>If checked (the default), the software deletes the data file and auxiliary files after successfully completing the load.</p> <p>If cleared or if the load does not complete successfully, the data file and auxiliary files remain in the bulk loader directory.</p>
Error Handling	
Constraints	The error handling table lets you select which types of constraint violations to ignore and whether to log the errors for each type. Constraint types include <i>Unique</i> , <i>Null</i> , <i>Data value</i> , <i>Foreign key</i> , and <i>All</i> . For <i>All</i> , type the maximum number of constraint violations the software ignores before stopping the job.
Ignore errors	Type the maximum number of violations the software encounters for any given constraint before stopping the job. Typing 0 means tolerate all errors of that constraint (allows unlimited errors).
Log errors	Select the type(s) of constraint violations to log in the message and row log files.

Related Information

[Common target table options \[page 256\]](#)

5.3.25.4.13 SAP Vora target table options

Use an SAP Vora datastore as a target in a data flow.

The following table contains options and descriptions specific for configuring an SAP Vora datastore as a target in a data flow. Find all other option descriptions for target tables in the topic “Common target table options.”

Option tab

Option	Description
Drop and re-create table	Selected by default. This option is required for SAP Vora.
Table type	<p>Select In-memory (default) or Disk.</p> <ul style="list-style-type: none">• In-memory: Performs bulk loading. Uses the relational in-memory store in SAP Vora. Loads relational data into the main memory for fast access. If you choose this option, you must configure a valid file location supported by SAP Vora. See the HDSF File Location option in the Bulk Loader Options table directly below.• Disk: Performs regular table loading. Uses the disk engine in SAP Vora. Provides relational data processing for data sets that do not fit into main memory. <p>Read about the engines in the SAP Vora <i>Installation and Administration Guide</i>.</p>

If you choose [In-memory](#) for the [Table type](#), the following options become available.

Bulk Loader Options

Option	Description
Bulk Load	<p>Uses SAP Vora bulk loading options to write data.</p> <div>Note Set the number of loaders in the General Advanced settings.</div>
HDFS File Location	Select the location of the local file, HDFS, or S3 HDFS to use for loading SAP Vora tables. If you use the bulk loader, the local file system should be your bulk loader directory.
Clean up bulk loader directory after load	<p>Selected by default. Select so the software deletes the data file and auxiliary files after successfully completing the load.</p> <p>If the load does not complete successfully, the data file and auxiliary files remain in the bulk loader directory for you to manually clean up. If you do not select bulk loading, the files remain in the local file system for you to manually clean up.</p>

Option	Description
Number of loaders	Default is 1 . When you enter more than 1, it is called parallel loading. For parallel loading, each loader receives the number of rows indicated in the Rows per commit option in turn. Then the software applies the rows in parallel with the other loaders.

Related Information

[SAP Vora datastore \[page 121\]](#)

[Common target table options \[page 256\]](#)

[General settings \[page 260\]](#)

[SAP Vora data type conversions \[page 341\]](#)

5.3.25.4.14 Teradata target table options

The following table contains option and description information specific to the [Bulk Loader Options](#) for Teradata target tables. All other option information for target tables can be found in the common target table options (see related links below).

On the [Bulk Loader Options](#) tab, the [Bulk loader](#) choices are:

- FastLoad
- MultiLoad
- TPump
- Parallel Transporter
- Load Utility
- None

The options on the [Bulk Loader Options](#) tab vary depending on the [Bulk loader](#) selected. However some options are common to all methods.

FastLoad, MultiLoad, TPump, and Parallel Transporter bulk loaders include several [Attributes](#). A different set of attributes displays depending on the bulk loader (and operator) selected. Attribute names in bold indicate that a value is required. You can accept the default values or modify them.

For details on all Teradata options and attributes, consult your Teradata documentation.

Common Teradata bulk loader options

Option	Description
File option	<p>Choose the type of file that will contain the data to bulk load:</p> <ul style="list-style-type: none"> • Data File • Generic Named Pipe • Named Pipes Access Module
<i>Data Services options</i>	
Generate files only	<p>When selected, the software generates a data file and a script file and ignores the <i>Number of loaders</i> option (on the <i>Options</i> tab). Rather than loading data into the target shown in the data flow, the software generates a control file and a data file that you can later load using Teradata bulk loading. This option is often useful when the target database is located on a system running a different operating system than that of the Job Server.</p> <p>The software writes the data and control files in the bulk loader directory (default value is <code><DS_COMMON_DIR>\log\bulkloader</code>) specified in the datastore definition. You must copy the files to the remote system manually. The naming conventions for these files is: <code><DatastoreName_OwnerName_TableName_n>.ctl</code> and <code><DatastoreName_OwnerName_TableName_n>.dat</code> where:</p> <ul style="list-style-type: none"> • <code><OwnerName></code> is the table owner • <code><TableName></code> is the target table • <code><n></code> is a positive integer, optionally used to guarantee that the software does not overwrite a pre-existing file
Clean up bulk loader directory after load	<p>Select to delete all bulk loader-related files (script, data files, temporary file) after the load is complete. If an error occurs during bulk loading, the software does not delete script and data files. Errors usually occur when:</p> <ul style="list-style-type: none"> • There is a syntax error in the script. • Error tables are not empty. Error tables contain rows that cannot be inserted into the target table due to data conversion or constraint violation.
Mode	<p>Specifies the mode for loading data into the target table:</p> <p><i>Append</i>: Adds new records to the table.</p> <p><i>Replace</i>: Deletes all existing records in the table, and then inserts the loaded data as new records.</p>
Field delimiter	Specifies a single-character field delimiter. Default value is /127 (non-printable character).
Bulk operation	<p>For MultiLoad and TPump and for the Parallel Transporter bulk loader Update and Stream operators, specify the bulk operation to use:</p> <p><i>Insert</i>: Insert rows.</p> <p><i>Upsert</i>: If the row exists, update it; if not, insert it.</p>
<i>Named pipes access module</i>	For bulk loaders FastLoad, MultiLoad, TPump, and Parallel Transporter with the <i>File Option Named pipes access module</i> selected, the following <i>Named pipe parameters</i> are available.

Option	Description
Named pipe parameters	<p>You can override the default settings for the following Teradata Access Module parameters:</p> <ul style="list-style-type: none"> • Logdirectory • Loglevel • Blocksize • FallbackFilename • FallbackDirectory • SignatureChecking <p>The Teradata Access Module creates a log file to record the load status. The Access Module log file differs from the tbuild log that you specify in the Log directory option. The Teradata Access Module writes information to fallback data files. If the job fails, the Teradata Access Module uses its log file and fallback data files to restart the load.</p> <p>The bulk loader directory is the default value for both Logdirectory and FallbackDirectory.</p> <p>For more information about these Access Module parameters, see your Teradata tools and utilities documentation.</p>

FastLoad bulk loader

FastLoad parameters	Description
Data encryption	Select to encrypt requests.
Print all requests	Allows printing of every request sent to the Teradata database.
Buffer size	<p>Output buffer size, in kilobytes used for Teradata FastLoad messages to the Teradata database.</p> <p>The output buffer size and the size of the rows in the Teradata FastLoad table determine the maximum number of rows that can be included in each message to the database. A larger buffer size reduces processing overhead by including more data in each message.</p> <p>The default buffer size is also the maximum size allowed: 63 kb. If a value greater than 63 kb is specified, Teradata FastLoad:</p> <ul style="list-style-type: none"> • Responds with a warning message • Resets the buffer size back to the default value • Continues with the Teradata FastLoad job
Character set	Character set specification for the target.

MultiLoad bulk loader

MultiLoad parameters	Description
Reduced print output	Select to limit the print output to the minimal information required to determine the success of the job.
Data encryption	Select to encrypt requests.

MultiLoad parameters	Description
Character set	Character set specification for the target.

TPump bulk loader

TPump parameters	Description
Reduced print output	Select to limit the print output to the minimal information required to determine the success of the job.
Retain macros	Select to keep macro(s) that TPump generates between jobs.
Data encryption	Select to encrypt requests.
Print all requests	Select to allow printing of every request sent to the Teradata database.
Number of buffers	Optionally increase the number of buffers per session from the default of 2 to a maximum of 10.
Periodicity value	This parameter is in effect whenever the Teradata BEGIN LOAD command uses the RATE parameter to control the rate at which statements are sent to the database. The default periodicity value is four 15-second periods per minute. To improve TPump work flow, adjust to a value from 1 to 30.
Character set	Character set specification for the target.
Configuration file	A file that contains various configuration and tuning parameters for TPump.

Parallel Transporter bulk loader

Option	Description
Operator	<p>Parallel Transporter operator values include:</p> <ul style="list-style-type: none"> • Load • Update • Stream • SQL Inserter <p>Note that the <i>Attributes</i> vary depending on the <i>Operator</i> selected. Refer to your Teradata documentation for information on attributes.</p>
Number of instances	Specify the number of instances for the load operator. This information is included in the Parallel Transporter script that Data Services generates.
Number of DataConnector instances	For the File Option <i>Data File</i> , specify the number of DataConnector instances for the read operator to read data files generated by Data Services. This information is included in the Parallel Transporter script that Data Services generates. The value should be less than or equal to the number of data files.

tbuild parameters

Debug all tasks	Enables debug trace functions for all tasks. Using this option outputs termination return codes that help with script debugging. Corresponds to the tbuild -d option.
Trace all tasks	Enables the trace option for all tasks. If not specified, trace is disabled. Corresponds to the tbuild -t option.

Option	Description
Latency interval (sec)	Latency is the interval value, in seconds, between the flushing of stale buffers. Corresponds to tbuild -l option.
Checkpoint interval (sec)	Specifies a time interval, in seconds, between checkpoints. Defaults to 10 seconds. Corresponds to tbuild -z option.
Load Utilities	
Option	Description
Command line	<p>Use this field to call a custom script. Load utilities include FastLoad, MultiLoad, and TPump. For example for FastLoad, you could enter:</p> <pre>fastload < C:\Teradata\FLScripts\myScript.ctl</pre> <p>Data Services does not parse or modify these scripts.</p>
Named pipe name	<p>For a Load Utility, if you choose either <i>Named Pipes Access Module</i> or <i>Generic Named Pipes</i> file option, enter the pipe name.</p> <p>On UNIX, the pipe is a FIFO (first in, first out) file that has name of this format: /temp/<filename>.dat On Windows, the pipe name has this format: \\.\pipe\<datastorename_ownername_tablename_loadernum>.dat</p>

Related Information

[Common target table options \[page 256\]](#)

[Teradata \[page 125\]](#)

[Teradata source \[page 217\]](#)

5.3.25.5 Common target table options

There are target table options that are common to all supported database types.

Set options in the following tabs:

- Target
- Options
- Load Triggers
- Pre Load Commands
- Post Load Commands

For more information about target tables, see “Embedded data flows” in the *Designer Guide*.

Related Information

[Target tab \[page 257\]](#)

[Options tab \[page 259\]](#)

[Target \[page 223\]](#)

5.3.25.5.1 Target tab

Descriptions of the options in the Target tab of the target table object editor.

To display target options, click the name of the target in the workspace or in the project area. This opens the object editor. The following table contains options in the Target tab of the object editor that are common to all supported database types.

Common options

Option	Description
Make port	Select this check box to make the target table an embedded data flow port.

Option	Description
Database type	<p>Select an item in the Database type box to set the content of additional tabs on the target table editor to match the specific options for that database type. This option allows you to quickly set target option values in data flows.</p> <p>If your target datastore has multiple configurations, all database types and their version numbers, which you specified in these configurations, are listed. To add or remove items in this list, edit the datastore configuration information in the datastore editor.</p> <p>The software allows you to use target table editor option values from any datastore configuration:</p> <ul style="list-style-type: none"> • If the datastore has only one configuration, then the initial values for the target table editor are defaults set by Designer for that database type or version. • If the datastore has more than one configuration and there are different database types/versions, then the software determines the initial values for the additional database types/versions from the Use values from box in the Create New Configuration dialog (a sub-dialog of the datastore editor). • If you also select the Restore values if they already exist check box (in the Create New Configuration dialog), SAP Data Services looks for previously defined values that once existed for that database type or version. It is possible for a data flow to contain target table editor values for a database type or version, even if its datastore configuration was deleted. The software retains all target table editor values saved with every datastore configuration. If such values exist, then it restores those values. Otherwise, it gets the values from the configuration you select from the Use values from option. For example, suppose you set a configuration for Oracle 8i. When you edit the target table editor options, you change the Rows Per Commit default value of 1000 to 500. Later you add a new datastore configuration for a Microsoft SQL Server 2000 database to your original datastore and set the Use values from option to Oracle 8i. In this case, the target table editor settings for SQL Server inherit the value 500 for Rows per Commit because this was the value set in the Oracle 8i configuration. <p>The values you set for the options in the target table editor are specific to the instance and database type/version of that object in the data flow. If you set values for one target table, any other target table in the same data flow is not affected.</p>
Database subtype	<p>If you choose Microsoft SQL Server 2016 for the Database type, select the database subtype from the drop-down menu, either Azure DW or APS.</p>

Related Information

[Common target table options \[page 256\]](#)

5.3.25.5.2 Options tab

Descriptions for target tables in the Options tab of the target table object editor.

To display options, click the name of the target in the workspace or in the project area. This opens the object editor. The Options tab is divided up into groups:

- General
- Error Handling
- Update Control
- Transaction Control

For Microsoft Azure DW and Microsoft APS, only the [Column comparison](#) and [Number of loaders](#) in the General section, the [Use overflow file](#) in the Error Handling section, and [Include in transaction](#) in the Transaction Control section are available. None of the options in the Update Control section are available.

The main Options tab contains the options described below:

Option	Description
Rows per commit	<p>Specifies the transaction size in number of rows.</p> <p>For example, if set to 1000, the software sends a commit to the underlying database every 1000 rows.</p> <p>Load triggers are never split across transaction boundaries, so if the load trigger crosses transaction boundaries, then the size of the transaction is automatically extended to accommodate the entire trigger requirement.</p> <p>For example, suppose you set Rows per commit to 3 and specified an insert trigger, where an incoming insert statement is converted into 5 statements. Rows per commit would automatically be extended to 5 to accommodate each insert trigger statement in a single transaction.</p> <p>This option is not available for targets in real time jobs.</p>
Delete data from table before loading	<p>For batch jobs, to clear the contents of the table before loading it, sends a TRUNCATE statement to databases that support it (Oracle, Microsoft SQL Server, SAP Sybase) or sends a DELETE statement to databases that do not support TRUNCATE. Default setting is not selected.</p> <p>For real-time jobs, clears data after processing each message. For real-time jobs you may want to deselect this option during development and testing.</p>

Option	Description
Drop and re-create table	Drops the existing table and creates a new one with the same name before loading. This option displays only for template tables. Template tables are used in design or test environments.
Table type This option is supported for DB2 10.1 or higher and Teradata.	Creates tables organized by row or column. Column Store: Create tables organized by column. Data types, blob , dbblob , and clob are not supported for this table type. Row store: Create tables organized by row.

Related Information

[Target tab \[page 257\]](#)
[General settings \[page 260\]](#)
[Error handling \[page 262\]](#)
[Update Control \[page 262\]](#)
[Transaction Control \[page 265\]](#)
[Load Triggers tab \[page 267\]](#)
[Common target table options \[page 256\]](#)

5.3.25.5.2.1 General settings

Read about the descriptions of the [General setting](#) in the [Options](#) tab of the target table object editor.

General options

Option	Description
Column comparison	<p>Specifies how the input columns are mapped to output columns. There are two options:</p> <p>Compare by position: The software disregards the column names and maps source columns to target columns by position.</p> <p>Compare by name: The software maps source columns to target columns by name.</p> <p>Validation errors occur if the data types of the columns do not match.</p>

Option	Description
Number of loaders	<p>Loading with one loader is known as “single loader loading.” Loading when the number of loaders is greater than one is known as “parallel loading.” The default number of loaders is 1. You can specify any number of loaders.</p> <p>When parallel loading, each loader receives the number of rows indicated in the Rows per commit option, in turn, and applies the rows in parallel with the other loaders.</p> <p>For example, if you choose a Rows per commit of 1000 and set the Number of Loaders to 3, the first 1000 rows are sent to the first loader. The second 1000 rows are sent to the second loader, the third 1000 rows to the third loader, and the next 1000 rows back to the first loader.</p>
Identity Upsert	<p>(Displayed only if the target table contains an SAP HANA Identity column. The default is Off.)</p> <p>This option works with both the normal SAP HANA loader and the bulk loader.</p> <p>When you turn this option On and you have mapped the Identity column, Data Services will generate the SQL statement to load the SAP HANA table containing the Identity column, and use the input value for insert, update, and upsert operations.</p> <p>If you do not map the Identity column, Data Services will issue an error.</p>
Enable Partitioning	<p>(Displayed only if the target table data is partitioned.)</p> <p>Loads data using the number of partitions in the table as the maximum number of parallel instances. You can select only one of the following loader options:</p> <ul style="list-style-type: none"> • Number of Loaders • Enable Partitioning • Transactional Loading <div> <p>i Note</p> <p>If you set Enable Partitioning to Yes and Include in transaction to Yes, the Include in transaction setting overrides the Enable Partitioning option.</p> </div>

Related Information

[Options tab \[page 259\]](#)

5.3.25.5.2.2 Error handling

Descriptions of *Error Handling* settings in the Options tab of the target table object editor.

Error Handling

Option	Description
<i>Use overflow file</i>	This option is used for recovery purposes. If a row cannot be loaded it is written to a file. When this option is set to Yes, options are enabled for the file name and file format. The default setting is No.
<i>File name</i> <i>File format</i>	These options are available only when you select Yes for the <i>Use overflow</i> file option. Specifies the file name and file format for the overflow file. The overflow format can include the data rejected and the operation being performed (write_data) or the SQL command used to produce the rejected operation (write_sql). You can enter a variable for the file name.

Related Information

[Options tab \[page 259\]](#)

5.3.25.5.2.3 Update Control

Descriptions of *Update Control* settings in the *Options* tab of the target table object editor.

Update control option descriptions

Option	Description
<i>Use input keys</i>	If the target table does not contain a primary key, this option enables the software to use the primary keys from the input. The default setting is <i>No</i> . If the target is a Microsoft SQL Server database table, and the identity column is mapped as the primary key, this option must = <i>No</i> .
<i>Update key columns</i>	This option is set to <i>No</i> by default. If you select <i>Yes</i> for this option, the software updates key column values when it loads data to the target.

Option	Description
<p><i>Auto correct load</i></p> <div> <p>i Note</p> <p>The optimizer automatically uses MERGE statements to improve auto correct load performance.</p> </div>	<p>Select <i>Yes</i> to use auto correct loading. Auto correct loading ensures that the same row is not duplicated in a target table. This is particularly useful for data recovery operations. The default setting is <i>No</i>.</p> <div> <p>i Note</p> <p>This option is not available for targets in real time jobs or target tables that contain LONG column(s).</p> </div> <p>When you select <i>Yes</i> for this option, the software reads a row from the source, then checks if the row exists in the target table with the same values in the primary key. If <i>Use input keys</i> is set to <i>Yes</i>, the software uses the primary key of the source table. Otherwise, the software uses the primary key of the target table; if the target table has no primary key, the software considers the primary key to be all the columns in the target.</p> <p>If a matching row does not exist, a new row is inserted, regardless of other options.</p> <p>If a matching row exists, the row is updated depending on the values of <i>Ignore columns with value</i>, and <i>Ignore columns with null</i>:</p> <ul style="list-style-type: none"> • When the column data from the source matches the value in <i>Ignore columns with value</i>, the corresponding column in the target table is not updated. The value may be spaces. Otherwise, the corresponding column in the target is updated with the source data. • When the <i>Ignore columns with null</i> option is set to <i>Yes</i> and the column data from the source is NULL, then the corresponding column in the target table is not updated. Otherwise, the corresponding target column is updated as NULL since the source column is NULL. <p>For supported databases, when the <i>Allow merge or upsert</i> option is enabled, the software can optimize data flows such that the database completes the auto correct load operation. When all other operations in the data flow can be pushed down to the source database, the auto-correct loading operation is also pushed down. The generated SQL implements the <i>Ignore columns with value</i> value if completed in the target editor, and the <i>Ignore columns with null</i> Yes/No setting.</p>

Option	Description
<p><i>Allow merge or upsert</i></p> <p>(DB2, Teradata, and Microsoft SQL Server 2008 and higher only)</p> <div> <p>i Note</p> <p>The optimizer automatically uses MERG statement for HP Vertica when user chooses Yes for <i>Auto correct load</i>.</p> </div>	<p>Specifies whether the Optimizer may use a MERGE statement to improve the performance of auto correct load functionality.</p> <p>Yes: Allows the Optimizer to consider using a MERGE statement during an auto correct load operation.</p> <p>If the Optimizer does not use a MERGE statement, it uses a T-SQL block to identify, insert, and update rows.</p> <p>No: The Optimizer will not use a MERGE statement to improve auto correct load performance.</p> <p>The default is Yes.</p> <div> <p>i Note</p> <p>When the data flow contains a Query transform with an Order by clause, the Optimizer always uses a T-SQL block to identify, insert, and update rows.</p> </div>
<p><i>Ignore columns with value</i></p>	<p>Enter a value that might appear in a source column and that you do not want updated in the target table. The value must be a string, it can include spaces, but the string cannot be in single or double quotations.</p> <p>When this value appears in the source column, the corresponding target column is not updated during auto correct loading.</p>
<p><i>Ignore columns with null</i></p>	<p>Select Yes if you do not want NULL source columns updated in the target table during auto correct loading.</p> <p>This option is only available when you select Yes for the <i>Auto correct load</i> option.</p>

Related Information




[Options tab \[page 259\]](#)

5.3.25.5.2.4 Transaction Control

Descriptions of *Transaction Control* settings in the Options tab of the target table object editor.

Transaction Control options

Option	Description
<i>Include in transaction</i>	<p>Indicates that this target is included in the transaction processed by a batch or real-time job. This option allows you to commit data to multiple tables as part of the same transaction. If loading fails for any one of the tables, no data is committed to any of the tables. The tables must be from the same datastore.</p> <p>Transactional loading can require rows to be buffered to ensure the correct load order. If the data being buffered is larger than the virtual memory available, the software reports a memory error.</p> <p>If you choose to enable transactional loading, the following options are not available:</p> <ul style="list-style-type: none">• Rows per commit• Use overflow file, and overflow file specification• Number of loaders• Enable partitioning• Bulk loader options• Pre load commands• Post load commands• Delete data from table before loading• Identity insert <p>The software does not push down a complete operation to the database if transactional loading is enabled.</p>
<i>Transaction order</i>	<p>Transaction order indicates where this table falls in the loading order of the tables being loaded. By default, there is no ordering. All loaders have a transaction order of zero. If you specify orders among the tables, the loading operations are applied according to the order. Tables with the same transaction order are loaded together. Tables with a transaction order of zero are loaded at the discretion of the data flow process.</p> <p>In Data flow view, the specified transaction order number appears nearer to the corresponding target.</p>

Option	Description
<i>Commit at end of INSERT...SELECT</i>	<p>When set to Yes (default), the software commits a single transaction log statement per load. When set to No, transaction size is limited to the value specified in <i>Rows per commit</i>. <i>If your transaction log size is too small for a single transaction of this type, set a commit size when the following applies:</i> to create a smaller transaction size than the single commit per job default behavior.</p> <ul style="list-style-type: none"> • If your transaction log size is too small for a singleThe job has a source and target that use the same data-store. • The job has an Oracle target table. • The software is optimizing and executing the job by pushing down the read operation to the Oracle target table host. This type of operation requires that the software generate an INSERT...SELECT SQL statement. The software commits a transaction for an INSERT...SELECT by default at the end of the job. • The job failed with an Oracle transaction log full error. <p>Troubleshooting:</p> <ol style="list-style-type: none"> 1. View the SQL that the software generates to see if an INSERT...SELECT statement is in use. (In the Designer, open a data flow and select  <i>Validate</i>  <i>Display Optimized SQL</i> .) 2. If an INSET...SELECT statement is in use, set the <i>Commit at end of INSERT...SELECT</i> to <i>No</i> and enter a value for <i>Rows per commit</i>. <p>If you use this option, expect to see a decrease in performance.</p>

Option	Description
Use NVARCHAR for VARCHAR columns in supported databases	<p>The software creates NVARCHAR columns in the template table for all VARCHAR columns in the input schema of the data flow. The data type displays as VARCHAR in the Designer, and when supported by the DBMS, displays as NVARCHAR in the database table.</p> <p>The following database management systems do not support the NVARCHAR data type:</p> <ul style="list-style-type: none"> • DB2 (non-UTF-8) • Oracle 8.x • Informix • SAP ASE • SAP Sybase IQ <p>For these systems, the software creates columns with VARCHAR data types and increases the column size using a codepage conversion factor based on the client code page defined in the datastore.</p> <div> <p>⚠ Caution</p> <p>Data loss may occur when transcoding from one national language to a different national language. Data loss will not occur when transcoding from a national language to Unicode.</p> </div> <div> <p>⚠ Caution</p> <p>Data truncation occurs when the column size of the source exceeds the maximum size allowed by the target DBMS.</p> </div>

Related Information

[Options tab \[page 259\]](#)

5.3.25.5.3 Load Triggers tab

Specifies SQL commands performed by the database on an INSERT, UPDATE, or DELETE operation.

You can specify a `<load trigger >` (a template SQL statement) that has placeholders for column and variable values. The software sets the placeholders at execution time based on the fields in the transform's input schema. For each row, the template is filled out and applied against the target.

i Note

Load trigger options are not supported for Microsoft APS or Azure DW.

The special operations you specify in a load trigger can occur before, after, or instead of normal operations.

Use load triggers in situations such as archiving updates to a warehouse or incremental updates of aggregation value.

The software does not parse load triggers. Thus, when you specify a load trigger, the software does not parameterize SQL statements. As a result, load times might be higher when you use load triggers.

The software does not validate load triggers.

i Note

If you use an override, you cannot specify auto correct load.

For example, instead of applying an insert of a new sales order rows, you use a load trigger that applies inserts and updates of aggregated values of sales_per_customer and sales_per_region.

The templates give you a row with customer_id, order_amount, region_id, and so forth.

The INSERT and UPDATE statements are:

```
INSERT into order_fact
values ([customer_id], [order_amount]);
UPDATE region_fact
SET order_amount =
order_amount + [order_amount]
WHERE region_id = [region_id];
```

Enter your load triggers manually or drag column names from the input schema. Column names must be enclosed in curly braces or square brackets. For example, {SalesOffice} or [SalesOffice].

With curly braces, the software encloses the value in quotation marks, if needed. With square brackets, it will not. To avoid unintended results, use curly braces for varchar or char column names.

If you insert column names into the SQL statement by dragging the column names, the software inserts square brackets for you. If you require curly braces, you must make the change from square brackets to curly braces.

[#insert], [#update], and [#delete] represent the default operations.

To delimit a SQL statement, use [#new]. For example:

```
[#insert] [#new]
insert into foo values ([col1], {col2}, ...)
```

For UPDATE operations you must specify both the "before" and the "after" image values. You can specify both images for INSERT and DELETE operations, also, but it is not required.

To specify "before" images, add the suffix .before to the column name. To specify "after" images, add the suffix .after to the column name.

The default suffix for UPDATE and INSERT operations is .after. The default suffix for DELETE operations is .before.

You can include variables in the SQL statements, but not expressions.

You can map a batch of SQL statements. Each SQL statement is separated by a new separator ([#new]).

The following statement is an example for mapping insert SQL:

```
INSERT into log_table values ({col1}, {col2})  
[#new]  
[#insert] [#new]  
delete from alt_junk where . . .
```

Related Information

[Options tab \[page 259\]](#)

[Common target table options \[page 256\]](#)

5.3.25.6 Target Data_Transfer files and tables

When you add a Data_Transfer transform to a data flow, you create a target for the temporary storage that SAP Data Services uses to process large amounts of data.

5.3.25.7 Target JSON files, messages, and templates

A JSON schema can be added to a job as a target.

Choose [Make File Target](#) or [Make Message Target](#) from the context menu that opens when you drop either format into the workspace.

You can also create a JSON file target without creating a format by using a Nested Schema template. The following table contains options from the Target tab of the [Nested Schemas Target File Editor](#) or the [Nested Schemas Target Message Editor](#) window.

Option	Description
Make port	Select this check box to make the target file an embedded data flow port.
File Location	(Optional) Select the name of a file location object. The file location object contains file transfer protocol information and local and remote server information. The software uses this information in target mode to safely transfer a copy of the generated output file from a local server to a remote server.

Option	Description
<i>Delete file after transfer</i>	<p>Available when you select a file location object for the <i>File Location</i> option above.</p> <ul style="list-style-type: none"> • Yes: Deletes the contents of the local server file after the software transfers the generated output data from the local file to the remote file. • No: Saves the contents of the local server file after the software transfers the generated output data from the local file to the remote file.
<i>File</i>	<p>For JSON File targets only.</p> <p>The location relative to the Job Server of a file to use as the target data file. If the file does not exist, the software creates it. If the file exists, the software overwrites the file with the output from the data flow.</p> <div> <p>i Note</p> <p>If your Job Server is on a different computer than the Designer, you cannot choose <code><Select file></code> to browse for the file path. In this case, type either an absolute path or a relative path that the Job Server can access. You can alternatively choose a variable.</p> </div>
<i>Delete and re-create file</i>	<p>Allows you to override the default behavior, which is to append new data to the target file. If selected, the software deletes the target file and creates a new one containing only the current data set.</p> <p>If you choose a file location object for the <i>File Location</i> option above, this setting affects the file in the remote server. The software outputs data to the local data file and then securely transfers the data to the remote data file.</p> <ul style="list-style-type: none"> • Checked: Applicable for FTP, SFTP, and SCP. Overwrites existing data in the remote server file with generated output data from the local server. • Unchecked: Not applicable for SCP and Local. Appends generated output data from the local file to an existing file in the remote server. <div> <p>i Note</p> <p>For all file transfer protocols except Local, if the file doesn't already exist in the remote server, the software creates a file and populates it with generated output data from the local server.</p> </div>
<i>Enable validation</i>	<p>Enable a comparison of the outgoing data to the stored Schema which this JSON file was created. When this option is enabled, the data flow throws an exception if the outgoing data is not valid. When you are developing real-time jobs, this validation helps you ensure sample data is both valid and well-formed. If you turn on this option in production, make sure to include appropriate error handling either in the job or the client application to process an error caused if the real-time job receives data that does not validate against the imported Schema.</p>

Option	Description
Test file	<p>(Message targets only)</p> <p>The location relative to the Job Server of a file to use as the message target when you run the job in test mode. If the file does not exist, the software creates it. If the file exists, the software clears the content of the file before writing the output to it.</p> <div> <p>i Note</p> <p>If your Job Server is on a different computer than the Designer, you cannot use Browse to specify the file path. You must type the path. You can type an absolute path or a relative path, but the Job Server must be able to access it. A variable can also be used.</p> </div>
Test file type	<p>(Message targets only)</p> <p>Specifies whether the test file is an XML file or a JSON file.</p>

Related Information

[File location object \[page 171\]](#)

5.3.25.8 Target XML files, messages, and templates

An XML Schema or DTD format can be added to a job as a target.

Choose [Make XML File Target](#) or [Make XML Message Target](#) from the context menu that opens when you drop either format into the workspace.

You can also create an XML file target without creating a format by using an XML template. The following table contains options from the [Target](#) tab of the [Nested Schemas Target File Editor](#) or the [Nested Schemas Target Message Editor](#) window for XML files, messages, and templates.

Option	Description
Make port	Select this check box to make the target file an embedded data flow port.
File Location	<p>(Optional) Select the name of a file location object.</p> <p>The file location object contains file transfer protocol information (such as FTP or SFTP) and local and remote server information. The software uses this information for target mode to transfer a copy of the generated output file from a local server to a remote server.</p>

Option	Description
<i>Delete file after transfer</i>	<p>Available when you select a file location object.</p> <ul style="list-style-type: none"> • Yes: Deletes the contents of the local server file after the software transfers the generated output data from the local file to the remote file. • No: Saves the contents of the local server file after the software transfers the generated output data from the local file to the remote file.
<i>File</i>	<p>(File targets only)</p> <p>The location relative to the Job Server of a file to use as the target. If the file does not exist, SAP Data Services creates it. If the file exists, the software overwrites the file with the output data.</p> <div> <p>i Note</p> <p>If your Job Server is on a different computer than the Designer, you cannot choose <code><Select file></code> to browse for the file path. In this case, type either an absolute path or a relative path that the Job Server can access. You can alternatively choose a variable.</p> </div>
<i>XML test file</i>	<p>(Message targets only)</p> <p>The location relative to the Job Server of a file to use as the message target when you run the job in test mode. If the file does not exist, the software creates it. If the file exists, the software clears the content of the file before writing the output to it.</p> <div> <p>i Note</p> <p>If your Job Server is on a different computer than the Designer, you cannot use Browse to specify the file path. You must type the path. You can type an absolute path or a relative path, but the Job Server must be able to access it. A variable can also be used.</p> </div>
<i>Delete and re-create file</i>	<p>Allows you to override the default behavior, which is to append new data to the target file. If selected, the software deletes the target file and creates a new one containing only the current data set.</p> <p>If you choose a file location object for the <i>File Location</i> option above, this setting affects the file in the remote server. The software outputs data to the local data file and then securely transfers the data to the remote data file.</p> <ul style="list-style-type: none"> • Checked: Applicable for FTP, SFTP, and SCP. Overwrites existing data in the remote server file with generated output data from the local server. • Unchecked: Not applicable for SCP and Local. Appends generated output data from the local file to an existing file in the remote server. <div> <p>i Note</p> <p>For all file transfer protocols except Local, if the file doesn't already exist in the remote server, the software creates a file and populates it with generated output data from the local server.</p> </div>
<i>Print comment</i>	<p>Allows you to include or exclude a comment in the target file data that identifies the data as having been processed by the software.</p>

Option	Description
<i>Replace NULL or blank</i>	Allows you to specify a value that will replace NULL or blank values in element data. Select the check box, then enter a value in the <i>With</i> field.
<i>Enable validation</i>	Enable a comparison of the outgoing data to the stored XML Schema or DTD from which this XML file was created. When this option is enabled, the data flow throws an exception if the outgoing data is not valid. When you are developing real-time jobs, this validation helps you ensure sample data is both valid and well-formed. If you turn on this option in production, make sure to include appropriate error handling either in the job or the client application to process an error caused if the real-time job receives data that does not validate against the imported XML Schema or DTD.
<i>Include schema location</i>	Allows you to include or exclude the schema location in the target file data. This check box is selected by default. If you do not want to include the schema location in the XML output, clear this check box.
<i>Include DTD</i>	(For targets created from DTD formats only) The content of an XML target does not normally include the DTD format (which the software uses internally). If you want to add the DTD format to the target file or message, select this check box.
<i>XML encoding</i>	Select an XML encoding for the XML target file. If you do not select a value, the encoding in the XML header field is used. If that field is empty, UTF-8 is used. XML file targets can be saved with a different encoding/code page than the software's system locale. XML message source and target encodings default to UTF-8 and cannot be changed.
<i>XML header</i>	You can use a unique header for each file target. To use this option, you must first enter the header information you want to use for the target. Thereafter, you can edit it from this field. For example, if your header includes more information than the XML Schema version and the encoding, you may want to view and edit this information in the Designer. If you only need to change the XML encoding for this file target, use the XML encoding option instead of editing the header.
<i>DTD file in DOCTYPE</i>	(For targets created from DTD formats only) The content of an XML target does not normally include the DTD format (which the software uses internally). If you want to add a DOCTYPE element to the target file or message, that specifies a path to a DTD format enter the path here or use the Browse button to select one.
<i>Format Name</i>	(Read only) The name of the DTD or XML Schema format used in the Designer.
<i>Root element name</i>	(Read only) The name of the root element used in the DTD or XML Schema.
<i>Namespace</i>	(Read only) The name space used in the XML Schema.

The validation for an XML target allows columns and nested tables marked as optional in the output schema to not be present in the input schema. At run-time the XML target will handle missing columns appropriately.

i Note

When the target file format contains a connection to a file location object, the software uses the information in the file location object for file transfer protocol to move the output file from the local server to the defined remote server.

Related Information

[Nested Schemas template \[page 200\]](#)

[Locales and Multi-byte Functionality \[page 1378\]](#)

[File location object \[page 171\]](#)


5.3.26 Target Writer migrated from Data Quality



If you migrate a Data Quality repository to SAP Data Services and you have projects that contain database Writer transforms, the resulting SAP Data Services jobs will contain migrated Writer targets. A migrated Writer target contains the SQL statements from the Data Quality Writer transform options [Write_SQL_Statement](#), [Pre_Write_SQL_Statement](#), and [Post_Write_SQL_Statement](#).

For more information, see the *Data Services Upgrade Guide*.

5.3.27 Template table

Characteristic	Description
	Template table icon.
Class	Reusable
Access	<ul style="list-style-type: none">• To insert as a target, open a data flow diagram in the work space, click the template table icon in the tool palette, and click anywhere in the data flow.• To insert as a source, open the object library, click the Datastores tab, select the desired template table, and drag into the data flow.• To view options, click the name of the template table in the workspace or in the project area. This opens the object editor.

Characteristic	Description
Description	<p>Template tables are new tables you want to add to a database. You can use a template table one time as a target and multiple times as a source. You cannot use a template table in an ABAP data flow.</p> <p>A template table provides a quick way to add a new target table to a data flow. When you use a template table, you do not have to specify the table's schema or import the table's metadata. Instead, during job execution, SAP Data Services has the DBMS create the table with the schema defined by the data flow. After you create a template table as a target in one data flow, you can use the same template table as a source in any other data flow.</p> <p>Use template tables in the design and testing phases of your projects. You can modify the schema of the template table in the data flow where the table is used as a target. Any changes are automatically applied to any other instances of the template table. During the validation process, the software warns you of any errors, such as errors that result from changing the schema.</p> <p>Before you can use a template table as a source in a data flow design, the data flow where the template table was created as a target has to be valid and you have to save the data flow.</p> <p>Before executing any job where a template table is used as a source, you must execute the job where the template table is used as a target at least one time. If the template table is used as a target and a source in the same job, then the data flow where it is used as a target must be executed first.</p> <p>When running a job where a template table is used as a target, use care if the table already exists. If the <i>Drop and re-create table</i> option is selected in the template table editor (this is the default option), the software drops the existing table and creates a new one. If the <i>Drop and re-create table</i> option is not selected, the software attempts to load data in the existing table. In this case, the software generates run-time errors if the existing table schema does not match the schema generated in the data flow.</p>

Related Information

[Target \[page 223\]](#)

[Message processing \[page 206\]](#)

5.3.27.1 Template table options

When used as a target, the options available from the target editor for template tables are the same as those available for target tables with some exceptions.

Target tab



Option	Description
Table name	The name of the table. Can contain alpha, numeric, and underscores; cannot include blank spaces.
Schema name	If you are using Netezza 7.x, you may enter a schema name in the schema box to limit the template tables to a particular schema. If you leave the schema name blank, the template tables are limited to the default schema.

Options tab

Option	Description
Column comparison	Drops the existing table and creates a new one with the same name before loading.
Drop and re-create table	Drops an existing table with the same name before creating the table specified by the template table. When using template tables in real-time jobs, deselect this and the Delete data from table before loading option. These options are selected by default when you create a template table.

Use NVARCHAR for VARCHAR columns in supported databases

Creates nvarchar columns in the template table for all varchar columns in the input schema of the data flow. The data type displays as varchar in the Designer, and, when supported by the DBMS, as nvarchar in the database table.

If you are using an ODBC datastore to connect to Oracle, in the datastore editor for [ODBC Miscellaneous](#)  [NVARCHAR type name](#) , select **NVARCHAR2**. See also [ODBC \[page 108\]](#).

The following database management systems do not support the nvarchar data type:

- DB2 (non-UTF-8)
- Oracle 8.x
- Informix
- SAP Sybase SQL Anywhere
- SAP ASE
- SAP Sybase IQ

For these DBMSs, the software creates columns with varchar data types and increases the column size using a codepage conversion factor based on the client code page defined in the datastore.

Caution

Data loss may occur when transcoding from one national language to a different national language. Data loss will not occur when transcoding from a national language to Unicode.

Caution

Data truncation occurs when the column size of the source exceeds the maximum size allowed by the target DBMS.

Bulk Loader Options tab


Option	Description
Bulk load	Not available for template tables.

Load Triggers tab

Option	Description
On operation	Not available for template tables.

Before running production jobs, execute the job to load the target table if you have not already done so, then right-click the template table in the object library or in a data flow and select Import Table. The software creates the table in your database and imports it. All information about the table is marked as part of the database and you can make no further changes to the schema. You can now use the new table in expressions, functions, transform options, or for bulk loading. Other features, such as exporting an object, are available for template tables.

5.3.28 Transform

Characteristic	Description
	Transform icon.
Class	Reusable
Access	In the object library, click the Transforms tab.
Description	Transforms define your data transformation requirements. Transforms use the operation codes associated with each row of data read from a source. The descriptions of individual transforms indicate which operation codes the transforms ignore or use.

Related Information

[Transforms \[page 351\]](#)


5.3.28.1 Transform attributes

Transforms have the following built-in attributes:

Attribute	Description
Name	The name of the object. This name appears on the object in the object library and in the calls to the object.
Description	Your description of the transform. Description is not available for query transforms.

If you delete a user-defined transform from the object library, calls to the object are replaced with an icon indicating that the calls are no longer valid, and it is deleted from the project area.


5.3.29 Try

Characteristic	Description				
	Try icon.				
Class	Single-use				
Access	With a work flow diagram in the workspace, click the try icon in the tool palette.				
Description	<p>A try is part of a serial sequence called a try/catch block. Use a single try with each try/catch block; there can be more than one catch with a single try. The try/catch block allows you to specify alternative work flows if errors occur while SAP Data Services is executing a job. Try/catch blocks "catch" classes of errors, apply solutions that you provide, and continue execution.</p> <p>Do not reference output variables from a try/catch block in any subsequent steps if you plan on using the automatic recovery feature. Referencing such variables could alter the results during automatic recovery.</p> <p>Tries have the following attribute:</p> <table> <tr> <th>Attribute</th><th>Description</th></tr> <tr> <td>Name</td><td>The name of the object. This name appears on the object in the diagram.</td></tr> </table>	Attribute	Description	Name	The name of the object. This name appears on the object in the diagram.
Attribute	Description				
Name	The name of the object. This name appears on the object in the diagram.				

Related Information

[Catch \[page 35\]](#)


5.3.30 While loop

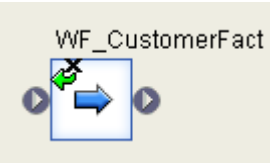
Characteristic	Description
	While loop icon.
Class	Reusable
Access	With a work flow diagram in the workspace, click the while loop icon in the tool palette.
Description	<p>A while loop repeats a sequence of steps as long as a condition is true.</p> <p>For each while loop, specify:</p> <ul style="list-style-type: none">• Condition: In the <i>While</i> box enter a Boolean expression that the job evaluates. The expression must evaluate to TRUE or FALSE. You can use constants, functions, variables, parameters, and standard operators to construct the expression.• Set of steps: In the while loop workspace, enter the steps you want completed when the condition is true. You can add any objects valid in a work flow, including scripts, work flows, and data flows. Connect these objects to represent the order that you want the steps completed. <div><p>i Note</p><p>Though you can include the parent work flow in the while loop, recursive calls can create an infinite loop.</p></div>

Related Information

[Smart editor \[page 299\]](#)

5.3.31 Work flow

Characteristic	Description
	Work flow icon.
Class	Reusable
Access	<ul style="list-style-type: none">• In the object library, click the <i>Work Flows</i> tab.• With a job or work flow diagram in the workspace, click the work flow icon in the tool palette.

Characteristic	Description
Description	<p>A work flow contains data flows and the operations that support data flows. The work flow defines the execution order of the data flows and supporting operations. A job is also a work flow.</p> <p>You can define parameters to pass values into the work flow. You can also define variables for use inside the work flow.</p> <p>In some cases, steps in a work flow depend on each other and should always be executed together. You can designate such a work flow (batch jobs only) as a "recovery unit." When designated as a recovery unit, the entire work flow must complete successfully during execution. If any step in such a work flow does not complete successfully, SAP Data Services re-executes all steps in the work flow during automatic recovery, except for ABAP data flows, the software re-executes data flows that executed successfully earlier. The software may or may not re-execute ABAP data flows.</p> <p>To designate a work flow as a recovery unit, Access work flow Properties, select <i>Regular</i> from the <i>Execution type</i> dropdown list and select the <i>Recover as a unit</i> check box.</p> <p>On the workspace diagram, a symbol indicates when a work flow is a recovery unit.</p> 

5.3.31.1 Work flow objects

The definition of a work flow can contain the following objects:

- Other work flows
- Data flows
- Scripts
- Try/catch blocks
- Conditionals
- While loops

5.3.31.2 Executing jobs only once

You can ensure that a job executes a work flow only one time by selecting *Regular* from the *Execution type* dropdown list and selecting the *Execute only once* check box on the data flow Properties window. When you select this check box, SAP Data Services executes only the first occurrence of the work flow and skips subsequent occurrences in the job. You might use this feature when developing complex jobs with multiple paths, such as those containing try/catch blocks or conditionals, and you want to ensure that the software executes a particular work flow only once.

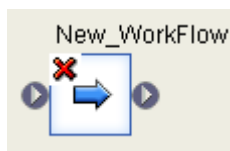
Before selecting the *Execute only once* option, note that:

- If you design a job to execute the same *Execute only once* work flow in parallel flows, the software only executes the first occurrence of the work flow, and you cannot control which one the software executes first.
Subsequent flows wait until the software processes the first one. The engine provides a wait message for each subsequent work flow. Since only one *Execute only once* work flow can execute in a single job, the engine skips subsequent work flows and generates a second trace message for each, "Work flow n did not run more than one time. It is an execute only once flow."
- If you design a job to execute more than one instance of the same *Execute only once* work flow, you must manage the values of output variables. The software only processes one such work flow per job. Subsequent instances of the work flow do not run and do not affect the values of variables in the job.
- The *Execute only once* work flow option does not override the *Enable recovery* job option.



Work flows have several built-in attributes.

Attribute	Description
Name	The name of the object. This name appears on the object in the object library and in the calls to the object.
Description	Your description of the work flow.
Date_created	The date that the object was created.

If you delete a work flow from the object library, calls to the object are replaced with an icon indicating that the calls are no longer valid.



5.3.32 XML file

Characteristic	Description
	XML file icons.
	
Class	Single-use

Characteristic	Description
Access	In the object library, click the Formats tab, then open the Nested Schemas category.
Description	<p>An XML file object allows you to indicate a source or target in a batch or real-time job. When used as a source, an XML file object translates incoming XML-formatted data into an internal SAP Data Services data set. When used as a target, an XML file object translates the data produced by a data flow, including nested data, into an XML-formatted file.</p> <p>The data read into or written out of an XML file must have a single row at the top-level table. When writing out an empty nested table, the software includes a single row of the nested table, with null values in each column of the table.</p> <p>To produce the metadata that describes the data that an XML file object handles, the software reads an XML Schema or DTD. The metadata is stored in the repository as either an XML Schema or DTD object.</p>

5.3.32.1 Source or target

You can insert an XML file into a data flow by dragging either an XML Schema or DTD format from the [Formats](#) tab of the object library into the workspace of a data flow. When you drop the format in the workspace, SAP Data Services prompts you to specify that the resulting object is a source or target file or a source or target message.

You can associate a file location object to an XML file source or target. The file location object contains specific file transfer protocol information to transfer XML files from a remote server to a local server (source) and a local server to a remote server (target).

Related Information

[File location object \[page 171\]](#)

[Target XML files, messages, and templates \[page 271\]](#)

5.3.32.2 Parsing input and producing output

Mapping rules govern how SAP Data Services translates an XML Schema or DTD into its internal schema definition and produces XML formatted data from an internal data set.

5.3.32.3 XML Properties

Context-click an XML nested schema in the [Formats](#) tab of the object library and select [Properties](#).

There are two tabs in the XML file Properties dialog: [General](#) and [Format](#).

General tab

Property	Description
<i>Name</i>	(Read only) The name of the XML schema file. This name appears in the object library under the Nested Schemas category of the Formats tab and is used for sources and targets (XML files or messages) that reference this format in data flows.
<i>Description</i>	Text that describes the XML. This text is entered when you create the DTD format file.



Format tab

Property	Description
<i>File Location</i>	<p>(Optional) Select the name of an existing file location object.</p> <p>The file location object contains file transfer protocol information (such as FTP or SFTP) and local and remote server information. The software uses this information in a data flow to copy the XML data from a remote to a local server (source) and from a local to a remote server (target).</p>
<i>Delete file after transfer</i>	<p>Available when you select a file location object above.</p> <p>Not applicable for SCP file transfer protocol. Check to delete the local file copy after the software loads it as a source in the data flow.</p> <p>Uncheck to save the local file copy after the software loads it as a source in the data flow.</p>
<i>File name</i>	<p>The location relative to the Job Server of an XML-formatted file to use as the source. You can enter a variable for this option.</p> <div data-bbox="821 1310 1399 1556"> <p>i Note</p> <p>If your Job Server is on a different computer than the Designer, you cannot choose <code><Select file></code> to use Browse to specify the file path. You must type the path. You can type an absolute path or a relative path, but the Job Server must be able to access it.</p> </div>
<i>Imported from</i>	(Read only) Contains the full path to the XML file.
<i>Root element name</i>	(Read only) The name of the primary node of the XML file. SAP Data Services only imports elements of the format that belong to this node or any sub nodes.
<i>Namespace</i>	(Read only) The Namespace name associated with the imported XML schema.

Related Information

[XML file source \[page 219\]](#)

5.3.33 XML message

Characteristic	Description
 XML Source	XML message icons.
 XML Target	
Class	Single-use
Access	In the object library, click the Formats tab, then open the Nested Schemas category.
Description	<p>An XML message object allows you to indicate a real-time source or target in a job.</p> <p>When used as a source, an XML message object translates incoming XML-formatted messages into an internal SAP Data Services data set. When used as a target, an XML message object translates the data produced by a job, including nested data, into an XML-formatted message and sends the message to the Access Server.</p> <p>When a real-time job contains an XML message source, it must also contain an XML message target.</p> <p>The data read into or written out of an XML message must have a single row at the top-level table. When writing out an empty nested table, the software includes a single row of the nested table, with null values in each column of the table.</p> <p>To produce the metadata that describes the data that an XML message handles, the software reads the format for the XML message. The metadata is stored in the repository as an XML Schema or DTD.</p>

Related Information

[DTD \[page 129\]](#)

[XML schema \[page 286\]](#)

5.3.33.1 Source or target

You can insert an XML message into a real-time job by dragging a XML Schema or DTD format from the [Formats](#) tab of the object library into the workspace of a data flow. When you drop the format in the workspace, you are prompted to specify that the resulting XML message as a source or target.

5.3.33.2 Source and Target editors

You can find information about source and target options elsewhere in the *Reference Guide*.

Related Information

[XML message source \[page 219\]](#)

[Target XML files, messages, and templates \[page 271\]](#)

5.3.33.3 XML test files

During the design phase of your application, you can execute a real-time job in "test mode." In test mode, the real-time job reads messages from an XML test file specified in the source editor, and writes XML-formatted messages to an XML test file specified in the target editor.

5.3.33.4 Parsing input and producing output

Mapping rules govern how SAP Data Services translates an XML Schema or DTD into its internal schema definition and produces XML from an internal data set.

See the Nested Data section of the *Designer Guide* for an introduction to the nested relational data model which the software uses to generate an internal hierarchical schema.

Related Information

[DTD \[page 129\]](#)

[XML schema \[page 286\]](#)

5.3.33.5 Properties



XML file properties are the same as those of its format.

Related Information

[DTD \[page 129\]](#)

[XML schema \[page 286\]](#)

5.3.34 XML schema

Characteristic	Description
	XML schema icon.
Class	Reusable
Access	In the object library, click the Formats tab, then open the Nested Schemas category.
Description	<p>SAP Data Services supports W3C XML Schemas Specification 1.0. This XML Schema version is documented on the following web site: www.w3.org/TR/2001/REC-xmlschema-1-20010502/ .</p> <p>XML Schemas describe the data structure of an XML file or message. Data flows can read and write data to messages or files based on a specified XML Schema format. You can use the same XML Schema to describe multiple XML sources or targets.</p> <p>To use XML Schemas, import XML Schema metadata into the software. During import, the software converts the structure defined in the XML Schema into the the software internal schema based on the nested relationship data model.</p>

Related Information

[Rules for importing XML Schemas \[page 292\]](#)

5.3.34.1 Editor

Open the XML Schema editor by double-clicking an XML Schema name in the object library.

```
<?xml version="1.0"?>
```



```

<xs:schema targetNamespace="http://my-company.com/namespace"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Order">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="OrderNo" type="xs:string" />
        <xs:element name="CustID" type="xs:string" />
        <xs:element name="ShipTo1" type="xs:string" />
        <xs:element maxOccurs="unbounded" name="LineItems">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Item" type="xs:string" />
              <xs:element name="ItemQty" type="xs:string" />
              <xs:element name="ItemPrice" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

The XML Schema editor displays:

- The nested schema. This example shows:
 - The object name is Simple_Order.
 - Columns at the top level are OrderNo, CustID, ShipTo1, and ShipTo2.
 - LineItems is a nested table.
 - Item, ItemQty, and ItemPrice are columns nested one level.
- The XML Format tab shows:
 - Full path to the XML Schema format file in the *Imported from* text box.
 - Root element in XML Schema format in the *Root element name* text box.
 - Namespace of XML Schema format in the *Namespace* text box.

5.3.34.2 Properties

XML schema property	Description
Name	The name of the format. This name appears in the object library under the Nested Schemas category of the Formats tab and is used for sources and targets (XML files or messages) that reference this format in data flows.
Description	Text that you enter to describe and document the XML Schema.
Imported from	The full path to the format. For example, D:\data\test.xsd.
Root element name	The name of the primary node you want to import. SAP Data Services only imports elements of the format that belong to this node or any sub nodes.
Namespace	(Optional) The Namespace URL of the root element.

5.3.34.3 Attributes supported for XML schemas

Column attribute	Description
Required	Indicates whether this column always has to be mapped (YES/NO)
Native Type	Original data type of the column. Saved as a string during import.
Default Value	Default value for this column.
Fixed Value	The only value the column can have.
Max Value Inclusive Of	Mapped from the following data type constraining facets: MaxInclusive, MinInclusive, MaxExclusive, and MinExclusive.
Min Value Inclusive Of	
Max Value Exclusive Of	
Min Value Exclusive Of	
Pattern	For a string: the pattern to which its value should match.
Enumeration	Contains a list of all possible values separated by vertical bars. For example: "Red White Blue Green Magenta". A string is cut off at 256 characters.
XML Type	Allows you to track whether the column was an element or attribute in the original XML Schema.
Length	Number of characters in this column.
Max Length	Maximum number of characters allowed in this column.
Min Length	Minimum number of characters allowed in this column.
Namespace	Column namespace.

Nested table attribute	Description
Any One Column	<p>If choice (for example, "white black almond"), then SAP Data Services sets the value of Any One Column to YES.</p> <p>If sequence (for example, "first, last, street, city, state") then the value is set to NO.</p> <p>If both are present in the XML Schema, the value is set to NO.</p>
Maximum Occurrence	<p>Indicates maximum number of rows in the table.</p> <p>If maximum occurrence equals zero, the software indicates that your XML Schema structure is not valid.</p>
Minimum Occurrence	Indicates minimum number of rows that can be in the table.
Optional Table	Indicates that an instance document might not contain this table, but the software still accepts the document as input.

Related Information

[Column attributes for tables \[page 222\]](#)

5.3.34.4 Supported XML Schema components

SAP BusinessObjects Data Services supports nearly all valid XML Schemas.

The software supports features such as abstract types and blocking in that the software will import and accept these features without error. Except for abstract types, this document does not discuss such features in depth as they do not have a direct impact on the ability of the software to support XML Schemas.

The software imports XML Schema data types as well as element and attribute names and their structure. Once imported, double-click an XML Schema format from the object library to view table and column names and structure. From the XML Format editor, right-click a column name and select edit properties, attributes, and data types.

Related Information

[Unsupported XML schemas \[page 293\]](#)

5.3.34.5 Abstract datatypes

When you build an XML file or message target using an XML schema that contains elements with abstract datatypes you must set the correct value for the "xsi:type" attribute to generate valid XML output. As data flow designer, you must know which of the many derived types is correct for any given element.

i Note

By default, all elements with abstract datatypes have an attribute called xsi:type.

When using XML Schemas with namespaces, you must include the right namespace in the type name. Obtain the right namespace tag by reviewing the namespace tags generated by SAP Data Services (typically ns1, ns2, ...) then using the tag that represents the right namespace in which the type exists.

Example:

Assume you have an element called Publication which has an abstract type called PublicationType. When the software imports this element, it will add an extra column called "xsi:type" as a child of Publication. You must then set the expression for this column to be equal to the expected type of the result (for instance, it could be BookType). To add the correct tag, first execute your job and note the generated tag names. For this example, it is ns1 for a namespace called <http://www.bookworld.com/>. So, for this example, the expression of xsi:type would be "ns1:BookType".

5.3.34.6 XML Schema elements

The following XML Schema elements are mapped to attributes when they are imported as metadata.

XML Schema Element	Attribute (nested table or column)
All	All. Elements should occur but they can occur in any order. See Choice.
Choice	<p>Any One Column.</p> <p>If the complex type for an element has been specified as choice then an attribute called Any One Column is created and set to YES.</p> <p>If the complex type has been defined with sequence or several nesting levels containing a mix of choice and sequence then the Any One Column table attribute is created and set to NO.</p> <p><code><Sequence></code>, <code><choice></code>, and <code><all></code> are handled as follows:</p> <ul style="list-style-type: none"> Sequence becomes "Any One Column = NO". Attributes A, B, and C become columns A, B, and C. Choice becomes "Any One Column = YES": Attributes A, B, and C become columns A or B or C. All becomes "All": B, C, and A or any combination of the three.
Default	Default Value
Enumeration	Enumeration. The value for this attribute is cut off after 256 characters. As a result, all the enumerated values may not be visible.
Fixed	Fixed Value
Length MinLength MaxLength	Length, Min Length, and Max Length.
MaxInclusive MinInclusive MinExclusive MaxExclusive	Max Value Inclusive Of, Min Value Inclusive Of, Min Value Exclusive Of, Max Value Exclusive Of.
MaxOccurs	Maximum Occurrence (only applies to tables).
MinOccurs	Minimum Occurrence (only applies to tables).
Name	Column name
Pattern	Pattern
TotalDigits and FractionDigits	None. Digits are handled as decimal data types. .
Type	Saved as the Native Type attribute (string). The Type element is also translated into a data type (usually <code>varchar</code>).
Sequence	See Choice.

Related Information

[Unsupported XML schemas \[page 293\]](#)

5.3.34.7 XML Schema attributes

The following XML Schema `<attributes>` are mapped to Data Services column attributes when they are imported as metadata.

XML Schema Attribute	Column Attributes
Default	Default Value
Fixed	Fixed Value The only value the column can have.
Name	Column name
Type	Saved as the Native Type attribute (string). The Type element is also translated into a data type (usually <code>varchar</code>).
Use	An XML Schema <code><Use></code> attribute with a value of OPTIONAL becomes the <code><Required></code> attribute with a value of NO. An XML Schema <code><Use></code> attribute with a value of REQUIRED becomes the <code><Required></code> attribute with a value of YES.

Related Information

[Unsupported XML schemas \[page 293\]](#)

5.3.34.8 Included XML Schemas

An XML Schema can be extended by including pointers to other XML Schema files. This is done by using `<import>`, `<include>` and `<redefine>`. These elements are defined at the schema level.

The difference between `<include>` and `<import>` is that for `<include>` the name spaces must be identical in both XML Schemas. `<Redefine>` is similar to `<include>` except the caller can redefine one or more components in the related XML Schema.

When you import an XML Schema, SAP Data Services follows the links to included files to define additional metadata. The included schema information is saved in the repository so that at run time there is no need to access these files again. Inclusions can be files or URLs.

5.3.34.9 Groups

XML Schemas allow you to group elements and then refer to the group. A similar concept is available for attributes (called an attribute group). In SAP Data Services any reference to a group will be replaced by the contents of that group.

5.3.34.10 Rules for importing XML Schemas

SAP Data Services applies the following rules to convert an XML Schema to the software's internal schema:

1. Any element that contains an element only and no attributes becomes a column.
2. Any element with attributes or other elements becomes a table.
3. An attribute becomes a column in the table corresponding to the element it supports.
4. Any occurrence of `<choice>`, `<sequence>` or `<all>` uses the ordering given in the XML Schema as the column ordering in the internal data set.
5. Any occurrence of `<maxOccurs,>` from greater than 1 to "unbounded", becomes a table with an internally generated name (an implicit table).
The internally generated name is the name of the parent followed by an underscore, then the string "nt" followed by a sequence number. The sequence number starts at 1 and increments by 1.

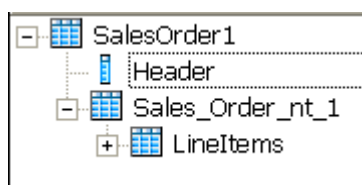
After applying these rules, the software uses two additional rules, except where doing so would allow more than one row for a root element:

1. If an implicit table contains one and only one nested table, then the implicit table can be eliminated and the nested table can be attached directly to the parent of the implicit table.

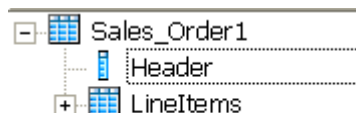
For example, the SalesOrder element might be defined as follows in an XML Schema:

```
<xs:element name="SalesOrder">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Header"/>
      <xs:element name="LineItems" minOccurs="0"
        maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

When converted in the software, the LineItems element with MaxOccurs = "unbounded" would become an implicit table under the SalesOrder table. The LineItems element itself would be a nested table under the implicit table.



Because the implicit table contains one and only one nested table, the format would remove the implicit table.

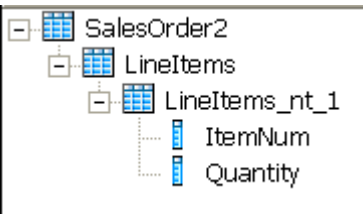


2. If a nested table contains one and only one implicit table, then the implicit table can be eliminated and its columns placed directly under the nested table.

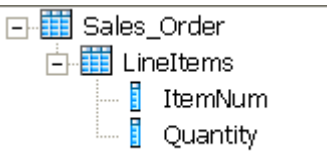
For example, the nested table Lineltems might be defined as follows in an XML Schema:

```
<xs:element name="SalesOrder">
  <xs:element name="LineItems" minOccurs="0"
maxOccurs="unbounded">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="ItemNum"/>
        <xs:element ref="Quantity"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

When converted into the software, the grouping with MaxOccurs ="unbounded" would become an implicit table under the Lineltems table. The ItemNum and Quantity elements would become columns under the implicit table.



Because the Lineltems nested table contained one and only one implicit table, the format would remove the implicit table.



5.3.34.11 Unsupported XML schemas

The following XML Schema elements and attributes are not supported in SAP Data Services. They are ignored and not imported.

Component	Description
Annotation	The softwares ignores both documentation and appinfo annotation components.
Non-native attributes	Non-native attributes are attributes that come from a name space other than the one your XML Schema uses. The W3C XML Schema standard enables users to add non-native attributes to attributes and elements. However, the software ignores all such attributes.

Component	Description
XDR files	The software does not support XML Data Reduced (XDR) files. XDR files were used as a format in some products before XML Schema became the standard. There are third-party tools on the market which can automatically convert XDR to XML Schema.

5.3.34.12 Limitations

If an XML schema definition contains the following elements or attributes, SAP Data Services imports it with the following limitations:

- Any element or anyAttribute
You can import an XML schema that contains an Any element or anyAttribute or both, but the format that the software creates does not show the Any element or anyAttribute.
Consequently, the software ignores the content of the Any element or anyAttribute when it reads an XML instance document. When an element has type anyType, the software treats everything within it as a string and does not recognize the subelements within it.
- Mixed content
The structure of an XML schema usually consists of elements that contain subelements, and the subelements at the lowest level contain character data. However, an XML schema definition allows character data to appear next to subelements, and the character data is not confined to the lowest level. For instance documents that contain mixed content, the software ignores the character data between any two subcolumns, but captures the values of the subcolumns.

5.3.34.13 Data type mappings

SAP Data Services imports data types for XML Schema elements and attributes.

There are two types of built-in data types, the Primitive data types and the Derived data types (derived from primitive). Each data type has the following values defined: space, lexical space, and constraining facet.

If the constraining facet `<length>` is missing when metadata is imported into the software, the default `varchar(1024)` is applied. Similarly, for a decimal the default values 28 and 2 are applied for `<precision>` and `<scale>`. All other facets like `<minInclusive>`, `<maxInclusive>`, `<minLength>` are imported as column attributes. Enumeration values are also imported as column attributes.

5.3.34.14 Primitive types

The following table lists Primitive XML Schema types, examples, and the corresponding data type in SAP Data Services. The constraining facets used are shown in bold.

XML Schema type	Example	Data type
AnyURI	http://www.example.com/	Varchar(len) : len = <i>length</i> (in chars)
Base64Binary	GpM7	Varchar(len) : len = <i>length</i> (in octets)
Boolean	{true, false, 0, 1}	Varchar(5)
Date	CCYY-MM-DD	Datetime
DateTime	Format = CCYY-MM-DD HH:MM:SS	Datetime
Decimal	7.999	Decimal(p, s) : p = <i>totalDigits</i> , a maximum of 28 and s = <i>fractionDigits</i> , default =28,2
Double	64 bit floating point	Double (In the software there is no difference between real and double)
Duration	P1Y2M3DT10H30M	Varchar(64)
Float	32 bit floating point, 12.78e-2	Real
gDay		Varchar(12)
gMonth		Varchar(12)
gMonthDay		Varchar(12)
gYear		Varchar(12)
gYearMonth	Gregorian CCYY-MM	Varchar(12)
HexBinary	0FB7	Varchar(len) : len = <i>length</i> (in octets)
Notation		N/A
Qname	po:USAddress	Varchar(len) : len = <i>length</i> (in chars)
String	"Hello World"	Varchar(len) : len = <i>length</i> (in characters)
Time	HH:MM:SS	Datetime

5.3.34.15 Derived types

The following table lists pre-defined Derived XML Schema types, examples, and the corresponding data type in SAP Data Services. The constraining facets used are shown in bold.

XML Schema type	Example	Data type
NormalizedString	[No tab/CR/LF in string]	Varchar(len) : len = <i>length</i> (in characters)
Token		Varchar(len) : len = <i>length</i> (in characters)
Language	En-GB, en-US, fr	Varchar(len): len = <i>length</i> (in characters)

XML Schema type	Example	Data type
NMTOKEN	US, Brésil	Varchar(len): len = <i>length</i> (in characters)
NMTOKENS	Brésil Canada Mexique	Varchar(len): len = <i>length</i> (in characters)
Name	ShipTo	Varchar(len): len = <i>length</i> (in characters)
NCName	USAddress	Varchar(len): len = <i>length</i> (in characters)
ID		Varchar(len): len = <i>length</i> (in characters)
IDREF		Varchar(len): len = <i>length</i> (in characters)
IDREFS		Varchar(len): len = <i>length</i> (in characters)
ENTITY		Varchar(len): len = <i>length</i> (in characters)
ENTITIES		Varchar(len): len = <i>length</i> (in characters)
Integer		Int
NonPositiveInteger		Int
NegativeInteger		Int
Long		Decimal 28,0
Int		Int
Short		Int
Byte		Int
NonNegativeInteger		Int
UnsignedLong		Long
UnsignedShort		Int
UnsignedByte		Int
PositiveInteger		Int
AnyType (ur-type)	unspecified type	Varchar(255)

5.3.34.16 User-defined types

User-defined types are XML Schema attributes with a non-XML Schema name space. The XML Schema W3C standard uses a SimpleType element for a user-defined type.

When SAP Data Services finds a user-defined type it finds the base type and uses it to assign a data type for the element. For example: If element X has type TelephoneNumber, its type in the software is `varchar(8)`.

Some simple types are based on other simple types. In such cases the software traces back to the base type.

5.3.34.17 List types

XML Schemas have list types. When it encounters a list, SAP Data Services makes that list's corresponding data type a `varchar(1024)`. All the elements of the list are placed in the value of that column as a string (exactly as it is represented in the XML).

5.3.34.18 Union types

A union type enables an attribute or element value to be one or more instances of one type drawn from the union of multiple primitive type and list types. When it encounters a union, SAP Data Services makes that union's corresponding data type a `varchar(1024)`.

5.3.34.19 Metadata

If you delete an XML Schema from the object library, XML sources or targets that are based on this format are invalid. SAP Data Services marks the source or target objects with an icon that indicates the calls are no longer valid.



To restore the invalid objects, you must delete the source or target and replace it with a source or target based on an existing XML Schema.

5.3.34.20 Error checking

SAP Data Services allows you to control whether it checks each incoming XML file or message for validity against the imported XML Schema. Select [Enable Validation](#) for an XML source or target in its editor. If you choose to check each XML file or message, the software uses the XML Schema imported and stored in the repository rather than a XML Schema specified by a given XML file or message. If a file or message is invalid relative to the XML Schema, the job produces an error and shuts down.

A typical example of when the software throws validation errors is when either a required element is missing or a new and unexpected element is present in the input. This is true of both the source and target. Consider the following examples:

- A element is defined in the XML Schema with enumeration values of "Black", "White", "StainlessSteel", and "Almond." If the mapping of that element from the XML document yields "Red" that would be incorrect

XML. But the XML file target generates the XML regardless. If validation is enabled, then this error is detected.

- If an element's whitespace attribute is set to `collapse`, the software does not change the data in sources or targets to respect this setting. The whitespace attribute is not supported in the software.

During development, you might validate all messages to test for error conditions with representative messages. During production, you might choose to accept rare invalid messages and risk ambiguous or incorrect data.

The software supports XML Schema legal naming such as allowing multiple elements and attributes to have the same name. However, name conflicts should be identified and tested before you import an XML Schema. The software cannot detect naming conflicts and may not report accurate errors which could later lead to runtime errors.

6 Smart editor

This section provides details about options available in the Designer smart editor. Use the smart editor to create scripts, expressions, and custom functions without having to type the names of existing elements like column, function, and variable names.

Related Information

[Scripting Language \[page 1259\]](#)

[Custom functions \[page 1242\]](#)

6.1 Accessing the smart editor

Access and use the embedded smart editor as a pane within any object editor in the Designer, or open the smart editor as a separate, full-size window.

For example, use the [Mapping](#) tab in a query to access smart editor:

1. Drag a column from an input schema into an output schema to enable the smart editor.
2. Enter text and select options using the smart editor's right-click menu, or click the ellipsis button to open the full-size smart editor window.

i Note

You cannot add comments to a mapping clause in a Query transform. For example, the following syntax is not supported on the Mapping tab:

```
table.column # comment
```

The job will not run and you cannot successfully export it. Use the object description or workspace annotation feature instead.

When you open the smart editor window, the context of the object from which you opened it is displayed in the title bar.

You can open the smart editor from the following locations:

- Query Editor [Mapping](#) tab
- Query Editor [From](#) tab
- Query Editor [Where](#) tab
- Script Editor
- Conditional Editor

- While Loop Editor
- Custom Function Editor
- Function wizard, "Define Input Parameter(s)" page
- SQL - Transform Editor
- Case Editor

6.2 Smart editor toolbar

In addition to standard toolbar icons (Open, Save, Print, Undo, Redo, Cut, Copy, Paste, Find, Replace, and Help), the smart editor toolbar also includes special icons to speed up your editing experience. Special smart editor toolbar icons include:

Icon	Description
Show/Hide Editor Library	Toggle to hide or show the Editor Library pane.
Show/Hide keypad	Toggle to hide or show an editing keypad.
Open selection list	Click to show a list of scripting options. Scroll and select the option you want. Double-click or press Enter/Return to add the option to your script.
Function wizard	Click to open the Function wizard window.
Validate	Click to check your script for errors.

6.3 Editor Library pane

Use the [Show/Hide Editor Library](#) icon in the toolbar to show or hide the smart editor library. The library:

- Displays functions, variables, and data using Tabs.
- Allows you to search each tab using the Find option.

6.3.1 Tabs

The [Functions](#) tab displays existing functions in SAP Data Services: built-in, custom, and imported.

The [Variables](#) tab displays variables, parameters, data type formats, and right-click menu options that can be used in the current context. For example, if you open the smart editor to create a custom function, the [Variables](#) tab will include options on its right-click menu that you can use to insert, delete, and define properties for new parameters and local variables.

The [Data](#) tab displays the schemas of data flow sources including nested schemas for the current context. For example, if you open the smart editor from an object in a data flow, such as a WHERE clause of a query,

schemas are displayed for connected sources. If you open the smart editor from a script object, the [Data](#) tab is not displayed.

6.3.2 Find option

You can search or browse through each tab to find the content you want to include in your script or expression.

6.4 Editor pane

6.4.1 Syntax coloring

When you type in the editor pane, the text changes color indicating the type of script language element it represents:

- Quoted strings are shown in pink
- Keywords in blue
- Comments in green
- Functions, operators, and variables are shown in black

6.4.2 Selection list and tool tips

You can use the smart editor with or without showing the editor library. The same list of context-based items available for use when the library is shown is also available in the editor when the selection list is enabled. The selection list shows these items in alphabetical order instead of grouping them into the categories shown in the library. In addition, the selection list displays keywords available for the context in which the editor is opened.

When the selection list is enabled, you can open it from the tool bar. The selection list also opens automatically when it recognizes a string pattern as you type into the editor.

6.4.3 Using the selection list and a tool tip

1. Right-click the editor, view the menu, and make sure that the [Enable Selection List](#) and [Enable Tool Tip](#) items are selected.
2. In the editor, enter at least three characters, or the dollar sign variable symbol (\$).

The selection list opens over the editor and highlights the first item (in this alphabetized list) that matches the characters you entered.

Alternatively, you can click the Open Selection list icon in the tool bar.

3. Double-click an item in the selection list to insert the item in the editor and view an associated tool tip.

The tool tip displays the same description, definition, or syntax that you would see if the item were selected from the editor library.

4. Complete the script or expression using these tools.

For example, if you are completing a look-up function, the tool tip will remain on the screen so that you can follow the syntax of the function. If you enter an input value of the wrong data type, the tool tip closes, indicating an error.

6.4.4 Right-click menu and toolbar

The right-click menu and the tool bar share many commands.

Menu	Toolbar	Key Command
	Open	
	Save As	
	Print	Ctrl + P
Undo	Undo	Ctrl + Z
Redo	Redo	Ctrl + Y
Cut	Cut	Ctrl + X
Copy	Copy	Ctrl + C
Paste	Paste	Ctrl + V
Select All		Ctrl + A
Find	Find	Ctrl + F
Replace	Replace	Ctrl + H
Validate	Validate	
Enable ToolTips		
Enable Selection List		
	Show/Hide Editor library	
	Show/Hide keypad	
	Open selection list	Alt+Down
	Function wizard	
	Help	

Important things to remember:

- [Enable ToolTips](#) and [Enable Selection List](#) can only be selected using the right-click menu.
- The library, key pad, selection list, and function wizard can be opened from the tool bar.
- Keyboard shortcuts are available for most commands.

6.5 Validating

Validate an expression, or validate the object that contains the expression.

If the software can validate the current context, such as an expression, it enables the *Validate* option in the tool bar and the right-click menu. If the software does not enable the *Validate* option, validate the expression in context of the whole object.

1. To validate an expression using the smart editor embedded error display:
 - a. Select the *Validate* icon in the tool bar or right-click and select *Validate*.



The software lists any errors in a separate pane below the editor.

- b. Double-click each error.

The editor redraws to show you where the error occurred in your text.

- c. Fix the error and continue validation until all errors are fixed and there are no more validation errors.
2. To validate an expression in context with the whole object:
 - a. Close the smart editor window.

The software displays the expression in the embedded smart editor.

- b. From the Designer menu, select **Debug > Validate**.

Related Information

[Debugging and Validation \[page 1267\]](#)

6.6 Browsing for a function

1. Expand the nodes to find the function you need:
 - Built-in functions are grouped by type
 - Custom functions are listed under the Custom node
 - Imported functions and stored procedures are listed under the name of the datastore used to import them.
2. Click a function and read its description and syntax in the yellow area below the tabs.
3. When you have the function you need, place it into the editor.

6.7 Searching for a function

1. Select the position in the editor where you want to place the function.
2. In the *Functions* tab of the editor library, select the *Find* node.
3. Enter a string such as `100`.
4. Press *Enter* or *Tab*.

All functions that contain the string are returned under the node.

5. To place the function into the editor, do one of the following:
 - double-click
 - drag-and-drop
 - right-click the function and select *Enter*
 - select the function and press *Enter* or *Tab*

7 Data Types

Data types are internal storage formats used to store values. A data type implies a default format for displaying and entering values. Expressions are a combination of constants, operators, functions, and variables that evaluate to a value of a given data type.

This section discusses how SAP Data Services processes, converts, and evaluates data types.

7.1 Descriptions of data types

Data types are internal storage formats used to store values. Data types also imply certain default formats for displaying and entering values.

SAP Data Services converts data based on whether the data is read or loaded:

- Data Services converts data read from sources to the appropriate Data Services data types.
- SAP Data Services converts data loaded to targets from SAP Data Services data types to types appropriate for the target.

The software recognizes the following data types:

- [date \[page 305\]](#)
- [datetime \[page 307\]](#)
- [decimal \[page 308\]](#)
- [double \[page 308\]](#)
- [int \(integer\) \[page 309\]](#)
- [interval \[page 309\]](#)
- [Limitations for long \[page 310\]](#)
- [numeric \[page 313\]](#)
- [real \[page 313\]](#)
- [time \[page 314\]](#)
- [timestamp \[page 315\]](#)
- [varchar \[page 316\]](#)

All of these data types allow NULL values.

7.1.1 date

The date data type defines calendar dates.

SAP Data Services automatically converts date values to and from the formats used by an external DBMS. Conversion operations from strings to dates, or from dates to strings or numbers require you to specify the

format of the date value. To specify a date format, generate a string from the following codes and other literal strings or punctuation.

Date format code	Description	Example
DD	2-digit day of the month value (1-31)	The 2nd day of the month: 02
MM	2-digit month number (1-12)	The month of March: 03
MONTH	Full name of the month	The first month of the year: JANUARY
MON	Abbreviated, three-character name of month	The first month of the year: JAN
YY	2-digit year A YY less than 15 is interpreted as being 20 <yy> ; for example, 10 would be interpreted as being the year 2010. A YY greater than or equal to 15 is interpreted as being 19 <yy> ; for example, 35 would be interpreted as being the year 1935. To change the value that the software uses to interpret 2-digit year dates, change the <i>Century Change Year</i> value in the Data options. (Select Tools > Options to open the <i>Options</i> window, and then select <i>General</i> under the <i>Data</i> category). The value must be a positive integer between 0 and 99.	The year 1998: 98
YYYY	4-digit year	The year 1999: 1999

You can perform various operations on dates such as add and subtract date, datetime, interval, and time values.

The following examples show the use of date formats with functions. The value of the variable `MyDate` is the first day of 1996.

i Note

If you use lower case to type “mon” or “month”, the resulting value of `to_char` will be in lower case (For example, jan or january). If you use upper case for “MON” or “MONTH”, the resulting value of `to_char` will be in upper case (For example, JAN or JANUARY).

Example	Output
<code>to_char(\$MyDate, 'YYYY.MM.DD')</code>	1996.01.01
<code>to_char(\$MyDate, 'MONTH DD, YYYY')</code>	JANUARY 01, 1996
<code>to_char(\$MyDate, 'DD/MM/YY')</code>	01/01/96
<code>to_date('01/01/96', 'DD/MM/YY')</code>	1996.01.01 stored as a date
<code>to_date('01/01/19', 'DD/MM/YY')</code>	2019.01.01 stored as a date

Related Information

[Date arithmetic \[page 317\]](#)

7.1.2 datetime

The datetime data type defines calendar dates and times.

SAP Data Services manages date operations in the format used by your DBMS. Conversion operations to or from datetime values require you to specify the format of the datetime. This data type behaves like a concatenation of two data items: The rules for the datetime type are the date rules for the date part, and the time rules for the time part.

If a date field is converted to a datetime value, the default time added to the value is 00:00:00. If a time is converted to a datetime value, the default date added to the value is 0000.01.01. You can also add and subtract date, datetime, interval, and time values.

When converting datetime values to strings, you can choose the sections of the value not to convert by excluding them from the format description. For example, to convert a datetime value to a string containing only the time, specify the function parameters as follows:

```
to_char($MyDateTime, 'hh24:mi:ss.ff')
```

For Oracle, if you load datetime data from SAP Data Services into a char field in an Oracle table, Oracle puts the data in its default datetime format—which includes only date values—and loses the time from the value.

Related Information

[Date arithmetic \[page 317\]](#)

7.1.3 decimal

The decimal data type defines exact decimal numbers.

When specifying a decimal data type in Data Services, you indicate the following characteristics of the type:

Precision: The total number of digits in the value.

Scale: The number of digits to the right of the decimal point.

In the software, the following relations must hold for precision and scale:

$1 \leq \text{precision} \leq 96$

$0 \leq \text{scale} \leq \text{precision}$

The decimal value can have a plus or minus sign indicating a positive or negative value. The sign can appear before or after the value with any number of blanks between the value and the sign. Unsigned values are considered to be nonnegative. The sign does not count as part of precision.

Leading zeros are permitted in the integer digit, and trailing zeros are permitted in the fraction part.

Input that is more precise than the data type of the column or variable in which it is stored is rounded. Input out of range (absolute value is too large) causes a runtime error.

i Note

Data Services uses a maximum of 28 precision. Data Services does not enforce precision (that is, having a larger number will not cause an error). Instead, Data Services will round any number more than 28.

The decimal data type and the numeric data type are identical in the software.

i Note

When you import a table from an Oracle datastore and the native column data type is NUMBER (without any specific precision and scale) the software imports the column as Decimal (28,7) by default. You can override the default Precision and Scale values for an Oracle table at the database level by entering the values in the Advanced section of the Create New DataStore option.

Related Information

[Create New Datastore: Oracle options \[page 109\]](#)

7.1.4 double

The double data type defines an 8-byte floating point value, with radix, exponent range, and precision of the platform on which SAP Data Services is running.

7.1.5 int (integer)

The `int` data type defines a 4-byte signed binary integer.

The `int` value can have a plus or minus sign indicating a positive or negative value. The sign can appear before or after the value with any number of blanks between the value and the sign. Unsigned values are considered to be nonnegative.

7.1.6 interval

The `interval` data type defines differences between dates or times. The value is in days unless you specify another unit, such as in a conversion function.

SAP Data Services provides conversion functions to make interval values accessible: `interval_to_char` and `num_to_interval`.

You can add and subtract date, datetime, interval, and time values.

This data type allows NULL values.

Related Information

[Date arithmetic \[page 317\]](#)

7.1.7 Limitations for long and blob

In general, you cannot use long or blob columns in comparisons, calculations, or data type conversions (except for `long_to_varchar` and `varchar_to_long`).

Therefore, you cannot use long or blob in the following situations:

- Join, key, compare, or pivot columns
- SQL functions, for example `substr`
- Expressions and conditions
- SELECT lists of queries containing GROUP BY clauses
- SELECT lists of queries with the Distinct Rows option enabled
- GROUP BY, ORDER BY, or WHERE clauses
- Input or output parameters or return type of functions; for example, `lookup`
- Variable data types
- Work flow and data flow input and output parameters
- Debug filters

The following table shows some of these limitations by transform.

Transform	Do not use long or blob column data in
Case	case condition
Hierarchy_Flattening	parent or child columns
History_Preserving	compare columns
Pivot	pivot transform
Table_Comparison	primary key columns or compare column
Query transform	WHERE, GROUP BY, ORDER BY, or DISTINCT

SAP Information Steward also ignores the auto-correct load option for target tables that contain a long or blob column. The software resets the option and issues a warning message at run time to indicate that the auto-correct load option has been disabled.

i Note

To use large object data types with Informix datastores, you must first configure the Informix ODBC options. For more information, refer to the Informix datastore options.

Related Information

[Informix \[page 93\]](#)

7.1.7.1 Limitations for long

SAP Data Services uses long to represent character-based large objects (clob).

Data Services also converts several other database specific large object types, such as longvarchar and text, to the long data type. The Data Services long data type supports mapping from all databases.

The software stores long columns either in memory or in the file system during the data flow execution depending on the size of the long value.

The following limitations exist for long data types:

- The software does not convert between long and any other data types except varchar. You can only convert long to or from varchar using the `varchar_to_long()` or `long_to_varchar()` functions.
- Long can be stored in blob.
- When loading a long or longraw column to an Oracle target, the software always extracts and loads the data in separate steps, so it cannot push down the SELECT and load operations in one statement. This restriction does not apply to the Oracle clob, nclob, or blob columns.

Long and blob data types share many of the same limitations.

Related Information

[Limitations for long and blob \[page 309\]](#)

[Limitations for blob \[page 311\]](#)

[Conversion to or from internal data types \[page 318\]](#)

7.1.7.1.1 File format considerations for long

You can define long data type columns in Data Services delimited files, XML files, and XML messages.

The long data can be in the file or can reference an external file. The notation for this external file is `<<filename>>`. The software automatically generates the file name.

For example, consider a comma-delimited file format that contains the following database columns.

Column name	Data type
ProductNo	integer
Description	long
Picture	blob

In Example 1, the long data appears in the file, but in Example 2, it references a file. The file name designates a path relative to the original input/output file or an absolute path.

Example 1:

```
7369,WidgetA transforms questionable data into trusted sources through a single environment.,<<pictures\WidgetA.jpg>>
```

Example 2:

```
7499,<<descriptions\WidgetC_descr.txt>>,<<C:\Widgets\pictures\WidgetC.jpg>>
```

7.1.7.2 Limitations for blob

SAP Data Services uses blob data types for multimedia data such as images, audio, and video.

The binary large object (blob) data type stores any kind of data in binary format. The software stores blob columns in the file system during the data flow execution.

The following limitations exist for blob data types:

- There are no data type conversions between blob and any other data types.
- Blob cannot be stored in long data type.
- Data Services does not support the NULL indicator for blob data.
- The Data Preview pane in the Designer cannot display blob data.

- The View Data utility shows the data for a blob column as <blob>.

Blob and long data types share many of the same limitations.

Related Information

- [Limitations for long \[page 310\]](#)
- [Conversion to or from internal data types \[page 318\]](#)
- [Limitations for long and blob \[page 309\]](#)

7.1.7.2.1 File format considerations for blob

You can define blob data type columns in SAP Data Services fixed-width files, delimited files, XML files, and XML messages.

You can define an unlimited number of blob columns in a file format, and the blob columns can appear in any order in a file format.

In fixed-width file formats:

- All blob columns are sized in bytes, not characters.
 - The minimum field size of a blob column is 1 byte.
 - The maximum field size is 32,768 bytes.
- Blob data is always inline with the rest of the data in the fixed-width file. The term inline means the data itself appears at the location where a specific column is expected.
For example, consider a fixed-width file format with the following columns.

Column name	Data type
EmployeeNo	integer(4)
LastName	varchar(15)
FirstName	varchar(15)
Password	blob(10)

In the following sample rows, the blob data (represented by '?'s) appears inline in the file.

7369	Washington	George	??????????
8272	Lincoln	Abraham	??????????

In delimited text files, XML files, and XML messages, blob columns always reference an external file. The notation for this external file is <filename>. The software automatically generates the file name.

For example, consider a comma-delimited file format that contains the following three columns.

Column name	Data type
ProductNo	integer
Description	long
Picture	blob

In the following sample row, the blob column references the external file.

```
7369,WidgetA transforms questionable data into trusted sources through a single
environment.,<<pictures\WidgetA.jpg>>
```

You can use blob data types in sources and targets and pass them through transforms.

- When the software reads blob data from fixed-width flat file, it does not trim the leading or trailing 0x00 bytes. It will not treat all 0x00s as Null. They will be stored as is.
- When the software loads a blob into a fixed-width flat file, if the size of the input blob data is not equal to the target blob field size, an error occurs. If the input blob consists of only 0x00s or is a NULL value, the software loads all 0x00s up to the size of the field size of the target blob column.

7.1.7.2.2 Database considerations for blob

Select the correct bulk loader when using blob with certain databases.

Data Services can load blob columns to:

- Databases that support parameterized loading.
- Databases with API-based bulk loaders (not file-based bulk loaders), which include:

Database	Bulk loader
Oracle	API
DB2	CLI load
Microsoft SQL Server	Bulk load
SAP ASE	Bulk load

7.1.8 numeric

The `decimal` data type and the `numeric` data type are identical in SAP Data Services. See [decimal \[page 308\]](#) for data type information.

7.1.9 real

The real data type defines a 4-byte floating point value, with radix, exponent range, and precision of the platform on which SAP Data Services is running.

The real value can have a plus or minus sign indicating a positive or negative value. The sign can appear before or after the value with any number of blanks between the value and the sign. Unsigned values are considered to be nonnegative.

Databases store real values as a 32-bit approximation of the number. Because of this approximation, comparison results are unpredictable when a real value is used in an equality or inequality comparison.

Therefore, it is recommended that you do not use a real value in a WHERE clause. Real values appear in WHERE clauses that the software generates when a column of type real is used:

- As a compare column in the Table_Comparison transform
- In the Map_Operation transform with an opcode of update or delete
- Explicitly in the WHERE clause of a Query transform

In some cases, columns of type real might unintentionally appear in the WHERE clause of these transforms. For example, when no compare columns are specified in a Table_Comparison transform, the transform uses all columns of the table as compare columns. Similarly, if the source of a Map_Operation transform does not have primary key specified and the opcode is update or delete, the transform uses all source columns in the WHERE clause of the UPDATE or DELETE statement.

Use caution when using the `real` data type in these transforms.

For more information, consult the appropriate reference material for Windows NT.

7.1.10 time

The time data type defines times of the day, with no calendar date.

SAP Data Services manages time operations in the format used by your database manager. Conversion operations to or from times require you to specify the format of the time value. To specify a time format, generate a string from the following codes and other literal strings or punctuation.

Time format code	Description
HH24	2-digit hour of the day (0-23)
MI	2-digit minute (0-59)
SS	2-digit second (0-59)

The following examples show the use of time formats with functions. The value of the variable `MyTime` is 25 minutes after 8 in the evening.

Example	Output
<code>to_char(\$MyTime, 'HH24:MI:SS.FF')</code>	20:25:00
<code>to_char(\$MyTime, 'HH24:MI')</code>	20:25

You can add and subtract date, datetime, interval, and time values.

Related Information

[Date arithmetic \[page 317\]](#)

7.1.11 timestamp

The `timestamp` data type supports the `timestamp` (with no zone information) data type in Oracle 9i. The `timestamp` data type incorporates up to a 9-digit sub-second.

Arithmetic

Add or subtract timestamp values. The resulting data type from addition or subtraction operations depends on the operation and data types involved. See [Date arithmetic \[page 317\]](#) for details.

Conversion between timestamp and character strings

You can convert between timestamp values and character values using the `to_date` and `to_char` functions. These functions have a format called `FF` which indicates the sub-second digits. For example, valid function calls are:

```
to_date ('2002.02.26 01234004 09:03:25','yyy.mm.dd ff hh24:mi:ss')
to_char (timestamp_column, 'yyyy.mm.dd hh24:mi:ss.ff')
```

Use the `FF` format for datetime columns to access sub-seconds. For example, a DB2 timestamp column is mapped to datetime in SAP Data Services. This column contains micro-second. You can access these sub-seconds using the `FF` format.

Limitations

You cannot use timestamp columns in the SQL transform or in an Oracle stored procedure.

To use a timestamp column in the SQL transform, convert the timestamp column in the select list of the SQL transform to a character format using the `to_char` function and convert it back to timestamp using the `to_date` function.

To use a timestamp column in an Oracle stored procedure, convert input and output timestamp parameters in the stored procedure to char, using the `to_char` function and convert the output parameter back to timestamp in SAP BusinessObjects Data Services using the `to_date` function. Alternatively, you can convert the input parameter back to timestamp in the stored procedure using the Oracle `to_timestamp` function.

7.1.12 varchar

When specifying a varchar data type, indicate the following characteristic of the type:

Characteristic	Description
Length	Number of characters that the variable or column can hold. Length must be greater than zero. There is no maximum allowable value for the length.

Character strings longer than the number of characters defined for the column or variable are truncated on the right to the length of the data type. Only the required number of characters is used to store strings shorter than length.

SAP Data Services provides functions to convert values to and from strings; to join strings together, use the concatenation operator (||). This data type allows NULL values.

The software conforms to the ANSI SQL-92 varchar standard and treats varchar data as follows:

- Keeps trailing blanks in character values that you insert into varchar columns.
- Keeps trailing blanks when you read from sources with string data types. If you want to remove trailing blanks from your input data, you must use the rtrim or rtrim_blanks function.
- Ignores trailing blanks when you compare varchar data in transforms (Case, Query, and Table_Comparison) and functions (decode, ifthenelse, lookup, lookup_ext, lookup_seq).

i Note

Not all database servers follow the ANSI standard for trailing blanks in insert, select, and compare operations. Therefore, if the software pushes down the insert, select, and compare operations to the database servers, the operations might return different results than when the software evaluates them. For the most current information on the treatment of trailing blanks, refer to the documentation for the specific database server.

- The ANSI standard treats an empty string as a zero length varchar value.

i Note

The software treats an empty string differently, depending on the source type. For example, Oracle treats an empty string as a NULL value, but DB2 and Microsoft SQL Server treat an empty string as a zero-length varchar value. For the most current information on the treatment of empty strings, refer to the documentation for your specific database server.

- When using the Equal (=) or Not Equal (<>) operators to compare a value with a NULL constant, the comparison always evaluates to FALSE. Use the IS NULL and IS NOT NULL operators to test for NULL values in the WHERE clause of the Query transform, the lookup_ext function, and the SAP Data Services scripting language.

If you currently run the software pre-version 11.5.0 scripts and data flows, it is recommended that you migrate them to use the ANSI varchar behavior because the previous varchar behavior will not be supported in a future version.

The software supports reading, transforming, and loading National Language Supported (NLS) data from different language locales using the varchar data type. The software supports national character-set data types in the following databases:

Database	Version	National character-set data type
DB2	7.0 and higher	graphic, vargraphic
MS SQL Server	7.0 and higher	nchar, nvarchar
Oracle	9i and higher	nchar, nvarchar2

When the software encounters a national character-set data type in an expression, it binds the column with the data type recommended by the database.

The engine reads and loads national character-set data types seamlessly without the need for you to configure a locale for a database client and its datastore for the columns that use these data types.

Related Information

[NULL values and empty strings \[page 1265\]](#)

[Processing with and without UTF-16 Unicode \[page 1384\]](#)

7.2 Data type conversion

This section discusses how Data Services processes various data types—conversions during arithmetic operations and between data types.

7.2.1 Date arithmetic

Data Services performs some implicit data type conversions on `date`, `time`, `datetime`, `timestamp`, and `interval` values when performing date arithmetic. The following table describes these conversions:

Operation	Return type
DATE + INTERVAL	DATE
TIME + INTERVAL	TIME
DATETIME + INTERVAL	DATETIME
INTERVAL + INTERVAL	INTERVAL
DATE - DATE	INTERVAL
DATE - INTERVAL	DATE
TIME - TIME	INTERVAL
TIME - INTERVAL	TIME
INTERVAL - INTERVAL	INTERVAL

Operation	Return type
DATETIME - DATE	INTERVAL
DATETIME - TIME	INTERVAL
DATETIME - DATETIME	INTERVAL
DATETIME - INTERVAL	DATETIME
TIMESTAMP + TIMESTAMP	INTERVAL
TIMESTAMP - TIMESTAMP	INTERVAL
TIMESTAMP + INTERVAL	TIMESTAMP
TIMESTAMP - INTERVAL	TIMESTAMP

7.2.2 Conversion to or from internal data types

Data Services performs data type conversions when it imports metadata from external sources or targets into the repository and when it loads data into an external table or file. The software uses its own conversion functions rather than those specific to the database or application that is the source of the data.

Additionally, if you use a template table or Data_Transfer table as a target, the software converts from internal data types to those of the respective DBMS.

7.2.2.1 Unsupported data types

Data Services can read, load, and invoke stored procedures involving unknown data types provided that your database servers can convert from VARCHAR to the native (unknown) data type and from the native (unknown) data type to VARCHAR. Data Services might have a problem loading VARCHAR to a physical CLOB column if the native database does not support that conversion (for example, bulk loading or auto-correct load could fail).

When the software encounters a column assigned to an unsupported data type, it does not import the metadata for the column and indicates an error. The file errorlog.txt contains an entry indicating the column that is ignored. To include the column in your job, convert the data type to one supported by the software before importing the metadata for the table.

i Note

Use the `varchar_to_long` function to convert a VARCHAR data type to a LONG datatype before loading physical CLOB. If from a prior installation, you are using a VARCHAR column in the physical schema for loading, this will still work.

7.2.2.2 Amazon Redshift data types

SAP Data Services converts Redshift data types to Data Services data types when Data Services imports metadata from a Redshift source or target into the repository.

The following table shows the data type conversions.

Redshift data type	Converts to Data Services data type
smallint	int
integer	int
bigint	decimal(19,0)
decimal	decimal
real	real
float	double
boolean	varchar(5)
char	char
<div><div>i Note</div><div>The char data type doesn't support multi-byte characters. The maximum range is 4096 bytes.</div></div>	
nchar	char
varchar	varchar
nvarchar	
<div><div>i Note</div><div>The varchar and nvarchar data types support utf8 multi-byte characters. The size is the number of bytes and the max range is 65535.</div></div> <div><div>⚠ Caution</div><div>If you try to load multi-byte characters into a char or nchar data type column, Redshift will produce an error. Redshift internally converts nchar and nvarchar data types to char and varchar. The char data type in Redshift doesn't support multi-byte characters. Use overflow to catch the unsupported data or, to avoid this problem, create a varchar column instead of using the char data type.</div></div>	
date	date
timestamp	datetime

Redshift data type	Converts to Data Services data type
text	varchar(256)
bpchar	char(256)

The following data type conversions apply when you create a template table:

Data Services data type	Redshift template table data type
blob	varchar(max)
date	date
datetime	datetime
decimal	decimal
double	double precision
int	integer
interval	float
long	varchar(8190)
real	float
time	varchar(25)
timestamp	datetime
varchar	varchar/nvarchar
char	char/nchar

7.2.2.3 Apache Hive

The following table shows the conversion between Apache Hive data types and Data Services data types when Data Services imports metadata from an Apache Hive source or target into the repository and when it loads data into an external table or file.

Hive data type	Converts to or from Data Services data type
TINYINT	INT

Hive data type	Converts to or from Data Services data type	
SMALLINT	INT	
INT/INTEGER	INT	
BIGINT	DECIMAL(19,0)	As default, the precision is 19.
FLOAT	DOUBLE	
DOUBLE	DOUBLE	
DECIMAL	DECIMAL	
VARCHAR	VARCHAR	
CHAR	VARCHAR	
STRING	VARCHAR(255)	
BOOLEAN	INT	
TIMESTAMP	DATETIME	
Date	Date	
INTERVAL	Not Supported	Only available starting with Hive 1.2.0
complex	Not Supported	Complex types are array, map, etc.

If a column is encountered that has an unsupported data type, Data Services will not import the column unless the property 'Import unsupported data types as VARCHAR of size' to the data store is specified.

7.2.2.4 Attunity Streams

The following table shows the conversion from Attunity Streams data types to Data Services data types when Data Services imports metadata from an Attunity Streams source or target into the repository.

Attunity Streams data type	Converts to Data Services data type	
bigint	double	
	<div> <i>i</i> Note <div>Because int is only four bytes, data is lost during the conversion.</div> </div>	
double	double	
float	double	

Attunity Streams data type	Converts to Data Services data type
long varchar	Converted to long only if imported using an ODBC datastore.
real	real
date	datetime
decimal	decimal
integer	int
smallint	int
time	time
timestamp	datetime
varchar	varchar

7.2.2.5 Cobol copybook

The following table shows the conversion from COBOL copybook data types to Data Services data types when Data Services imports metadata from a COBOL copybook source or target into the repository.

COBOL copybook data type	Converts to Data Services data type
computational	decimal
comp	decimal
computational-1	float
comp-1	float
computational-2	double
comp-2	double
computational-3	decimal
comp-3	decimal
computational-4	decimal
comp-4	decimal
computational-5	decimal
comp-5	decimal
computational-6	decimal
comp-6	decimal
computational-x	decimal
comp-x	decimal
binary	decimal
packed-decimal	decimal

COBOL copybook data type	Converts to Data Services data type
float	float
double	double
signed-short	int
unsigned-short	int
signed-int	int
unsigned-int	int
signed-long	int
unsigned-long	int
integer	int
DISPLAY, PICTURE contains A or X (character data)	varchar
Floating point like +9.9(5)E+99	double

7.2.2.6 Data Federator

The software converts data types from Data Federator when you import tables from Data Federator.

Data type conversions

Data Federator	Data Services
Decimal	Decimal precision and scale (28,6)
VarChar	VarChar (1024)

You may change the decimal precision or scale and varchar size within the software after importing from the Data Federator data source.

7.2.2.7 HP Vertica data type conversion

Data type conversion between HP Vertica and SAP Data Services.

HP Vertica data type	Data Services data type
Boolean	Int
Integer, INT, BIGINT, INT8, SMALLINT, TINYINT	Decimal
FLOAT	Double

HP Vertica data type	Data Services data type
Money	Decimal
Numeric	Decimal
Number	Decimal
Decimal	Decimal
Binary, Varbinary, Long Varbinary	Blob
Long Varchar	Long
Char	Varchar
Varchar	Varchar
Char(n), Varchar(n)	Varchar(n)
DATE	Date
TIMESTAMP	Datetime
TIMESTAMPTZ	Varchar
Time	Time
TIMETZ	Varchar
INTERVAL	Varchar

Data type conversion from internal data types to HP Vertica data types for template tables or Data_Transfer transform tables.

Data Services data type	HP Vertica data type in template table
Blob	Long Varbinary
Date	Date
Datetime	Timestamp
Decimal	Decimal
Double	Float
Int	Int
Interval	Float
Long	Long Varchar

Data Services data type	HP Vertica data type in template table
Real	Float
Time	Time
Varchar	Varchar
Timestamp	Timestamp

7.2.2.8 IBM DB2

The following table shows the conversion between DB2 data types and Data Services data types when Data Services imports metadata from a DB2 source or target into the repository and when it loads data into an external table or file.

DB2 data type	Converts to or from Data Services data type
bigint	int <div> i Note Because int is only four bytes, data is lost during the conversion to Data Services data type. </div>
blob	blob <div> i Note This data type is not supported when importing data to column-stored template tables. </div>
character	varchar
clob	long <div> i Note This data type is not supported when importing data to column-stored template tables. </div>
date	date
dbclob	long <div> i Note This data type is not supported when importing data to column-stored template tables. </div>
decimal	decimal

DB2 data type	Converts to or from Data Services data type
double	double
float	double
graphic	varchar
integer	int
long varchar	long
long vargraphic	long
real	real
smallint	int
time	time
timestamp	datetime
varchar	varchar
vargraphic	varchar

The following table shows the conversion from internal data types to DB2 data types in template tables or Data_Transfer transform tables.

Data Services data type	DB2 data type in template table
blob	blob
date	date
datetime	date
decimal	decimal
double	double
int	int
interval	real
long	clob
numeric	character
real	real
time	time
varchar	varchar
timestamp	date

7.2.2.9 Informix

The following table shows the conversion between Informix data types and Data Services data types when Data Services imports metadata from an Informix source or target into the repository and when it loads data into an external table or file.

Informix data type	Converts to or from Data Services data type
blob	blob
byte	blob
char	varchar
character	varchar
character varying	not supported
clob	long
date	date
datetime	datetime
dec	decimal
decimal	decimal
double	double
float	double
int	integer
integer	integer
money	decimal
numeric	decimal
real	real
serial	not supported
smallfloat	double
smallint	integer
text	long
varchar	varchar

The following table shows the conversion from internal data types to Informix data types in template tables or Data_Transfer transform tables.

Data Services data type	Informix data type in template table
blob	blob
date	date
datetime	date
decimal	decimal
double	float
int	int
interval	int
long	clob
numeric	decimal

Data Services data type	Informix data type in template table
real	real
time	date
varchar	varchar
timestamp	date

7.2.2.10 Microsoft Excel

Microsoft ActiveX Data Objects (ADO) makes it possible to format and convert Excel data sources. The following table shows the conversion from ADO data types to Data Services data types when Data Services imports metadata from an Excel source or target into the repository.

ADO data type	Converted to Data Services data type
adDouble	double
adCurrency	double
adBoolean	varchar
adDate	timestamp
adDBTimestamp	timestamp
ad...Char	varchar

7.2.2.11 Microsoft SQL Server

The following table shows the conversion between Microsoft SQL Server data types and Data Services data types when Data Services imports metadata from a Microsoft SQL Server source or target into the repository and when it loads data into an external table or file.

Microsoft SQL Server data type	Converts to or from Data Services data type
binary	not supported
bigint	decimal
bit	int
char	varchar
date (SQL Server 2008 and higher only)	date
datetime	datetime
datetime2 (SQL Server 2008 and higher only)	datetime

Microsoft SQL Server data type	Converts to or from Data Services data type
decimal	decimal
float	double
image	blob
int	int
money/smallmoney	decimal
nchar	varchar
ntext	long
numeric	decimal
nvarchar	varchar
nvarchar(max)	long
real	real
smalldatetime	datetime
smallint	int
text	long
time (SQL Server 2008 and higher only)	time
<div> <div></div> <div> i Note On UNIX, the sub-seconds (HH:MI:SS.<ff>, where ff represents the sub-seconds) for the time data type are always set to zero (0). To get a more precise time, use the datetime or datetime2 data type. </div> </div>	
timestamp	not supported
tinyint	int
uniqueidentifier	varchar
varbinary	not supported
varbinary(max)	blob
varchar	varchar
varchar(max)	long
xml	long

The following table shows the conversion from internal data types to Microsoft SQL Server data types in template tables or Data_Transfer transform tables.

Data Services data type	MS SQL Server data type in template table
blob	image
date	datetime
	date (SQL Server 2008 and higher only)

Data Services data type	MS SQL Server data type in template table
datetime	datetime datetime2 (SQL Server 2008 and higher only)
decimal	decimal
double	float
int	int
interval	real
long	text
numeric	decimal
real	real
time	datetime time (SQL Server 2008 and higher only)
varchar	varchar
timestamp	datetime datetime2 (SQL Server 2008 and higher only)

7.2.2.12 MySQL

The following table shows the conversion between MySQL data types and Data Services data types when Data Services imports metadata from a MySQL source or target into the repository and when it loads data into an external table or file.

MySQL data type	Converts to or from Data Services data type
bigint	decimal
decimal	decimal
dec	decimal
bit	int
tinyint	int
bool	int
smallint	int
mediumint	int
int	int
integer	int
year	int
float	real

MySQL data type	Converts to or from Data Services data type
double	double
datetime	datetime
timestamp	datetime
date	date
time	time
varchar	varchar
nvarchar	varchar
nchar	varchar
char	varchar
enum	varchar
set	varchar
tinytext	long
text	long
mediumtext	long
longtext	long
tinyblob	blob
blob	blob
mediumblob	blob
longblob	blob

The following table shows the conversion from internal data types to MySQL data types in template tables or Data_Transfer transform tables.

Data Services data type	MySQL data type in template table
blob	blob
date	date
datetime	timestamp
decimal	decimal
double	double
int	int
interval	float
long	text
numeric	decimal
real	float
time	time
varchar	varchar

Data Services data type	MySQL data type in template table
timestamp	timestamp

7.2.2.13 Netezza

The following table shows the conversion between Netezza data types and Data Services data types when Data Services imports metadata from a Netezza source or target into the repository and when it loads data into an external table or file.

Netezza data type	Converts to or from Data Services data type
bigint	decimal
boolean	int
byteint	int
char	varchar
date	date
double precision	double
float	double
integer	int
interval	varchar
nchar	varchar
nvarchar	varchar
numeric	decimal
real	real
smallint	int
time	time
time with time zone	varchar
timestamp	datetime
varchar	varchar

The following table shows the conversion from internal data types to Netezza data types in template tables or Data_Transfer transform tables.

Data Services data type	Netezza data type in template table
date	date
datetime	timestamp
decimal	numeric
double	double precision

Data Services data type	Netezza data type in template table
int	integer
interval	real
numeric	numeric
real	real
time	time
timestamp	timestamp
varchar	varchar

7.2.2.14 ODBC

The following table shows the conversion between ODBC data types and Data Services data types when Data Services imports metadata from an ODBC source or target into the repository and when it loads data into an external table or file.

ODBC data type	Converts to or from Data Services data type
bigint	decimal
char	varchar
datalink	not supported
date	date
decimal	decimal
double	double
float	double
graphic	not supported
int	int
nclob	not supported
numeric	decimal
real	real
sql_longvarchar	long
sql_longvarbinary	blob
sql_wlongvarchar	long
time	time
timestamp	datetime
tinyint	int
user-defined	not supported
varchar	varchar

The following table shows the conversion from internal data types to ODBC data types in template tables or Data_Transfer transform tables.

Data Services data type	ODBC data type in template table
blob	sql_long varbinary
date	date
datetime	timestamp
decimal	decimal
double	double
int	int
interval	real
long	sql_long varchar
numeric	decimal
real	real
time	time
varchar	varchar
timestamp	timestamp

7.2.2.15 Oracle

The following table shows the conversion between Oracle data types and Data Services data types when Data Services imports metadata from an Oracle source or target into the repository and when it loads data into an external table or file.

Oracle data type	Converts to or from SAP Data Services data type
char	varchar
blob	blob
clob	long
date	datetime
decimal	decimal
doubleprecision	double
float	double
label	not supported
long	long
long raw	blob
nchar	varchar
nclob	long

Oracle data type	Converts to or from SAP Data Services data type
number	int: If scale is 0 and precision is < 9 decimal: All other
nvarchar2	varchar
real	double
row	not supported
rowid	not supported
timestamp	timestamp
varchar	varchar
varchar2	varchar

The following table shows the conversion from internal data types to Oracle data types in template tables or Data_Transfer transform tables.

Data Services data type	Oracle data type in template table
blob	blob
date	date
datetime	date
decimal	decimal
double	double
int	int
interval	real
long	clob
numeric	number
real	double
time	date
varchar	varchar2
timestamp	timestamp

7.2.2.16 SAP HANA

Data type conversion when SAP Data Services imports metadata from an SAP HANA source or target into the repository and then loads data to an external table or file.

SAP HANA data type	Converts to Data Services data type
integer	int

SAP HANA data type	Converts to Data Services data type
tinyint	int
smallint	int
bigint	decimal
char	varchar
nchar	varchar
varchar	varchar
nvarchar	varchar
decimal or numeric	decimal
float	double
real	real
double	double
date	date
time	time
timestamp	datetime
clob	long
nclob	long
blob	blob
binary	blob
varbinary	blob

The following table shows the conversion from internal data types to SAP HANA data types in template tables.

Data Services data type	Converts to SAP HANA data type
blob	blob
date	date
datetime	timestamp
decimal	decimal
double	double
int	integer
interval	real
long	clob/nclob
real	decimal
time	time
timestamp	timestamp
varchar	varchar/nvarchar

7.2.2.17 SQL Anywhere

The following table shows the conversion between SAP Sybase SQL Anywhere data types and Data Services data types when Data Services imports metadata from a SQL Anywhere source or target into the repository and when it loads data into an external table or file.

SQL Anywhere data type	Converts to or from Data Services data type
BIT	int
[UNSIGNED] TINYINT	int
[UNSIGNED] SMALLINT	int
INT or INTEGER	int
UNSIGNED INT or INTEGER	decimal(10,0)
BIGINT	decimal(19,0)
UNSIGNED BIGINT	decimal(20,0)
DEC or DECIMAL	decimal
NUMERIC	decimal
FLOAT(1-24) or FLOAT	real
FLOAT(25-53)	double
REAL	real
DOUBLE [PRECISION]	double
SMALLMONEY	decimal(10,4)
MONEY	decimal(19,4)
CHAR	varchar
NCHAR	varchar
VARCHAR	varchar
NVARCHAR	varchar
LONG VARCHAR	long
LONG NVARCHAR	long
TEXT (LONG VARCHAR)	long
NTEXT (LONG NVARCHAR)	long
XML	long
UNIQUEIDENTIFIERSTR (CHAR(26))	varchar(36)
DATE	date
TIME	time
SMALLDATETIME	datetime
DATETIME	datetime
TIMESTAMP	datetime

SQL Anywhere data type	Converts to or from Data Services data type
BINARY	blob
IMAGE	blob
LONG BINARY	blob
VARBINARY	blob
UNIQUEIDENTIFIER	blob

The following table shows the conversion from internal data types to SQL Anywhere data types in template tables or Data_Transfer transform tables.

Data Services data type	SQL Anywhere data type in template table
blob	LONG BINARY
date	DATE
datetime	DATETIME
decimal	DECIMAL
double	DOUBLE
int	INTEGER
interval	REAL
long	LONG VARCHAR/LONG NVARCHAR
real	REAL
time	TIME
timestamp	DATETIME
varchar	VARCHAR/NVARCHAR

i Note

When using SQL Anywhere:

- TIMESTAMP supports six-digit sub-seconds (i.e. milliseconds)
- DECIMAL supports precision up to 96 due to ACTADECIMAL_MAXPRECISION (SQL Anywhere supports precision up to 127)
- VARCHAR columns support max length of 32767 characters
- LONG columns support a maximum length of 2GB characters

7.2.2.18 SAP ASE

The following table shows the conversion between SAP ASE data types and Data Services data types when Data Services imports metadata from a SAP ASE source or target into the repository and when it loads data into an external table or file.

SAP ASE data type	Converts to or from Data Services data type
binary	not supported
bit	int
char	varchar
datetime	datetime
decimal	decimal
double	double
float	double
image	blob
int	int
money	decimal(20,4)
numeric	decimal
real	real
smalldatetime	datetime
smallint	int
smallmoney	decimal(12,4)
text	long
timestamp	not supported
tinyint	int
varbinary	not supported
varchar	varchar

The following table shows the conversion from internal data types to SAP ASE data types in template tables or Data_Transfer transform tables.

Data Services data type	SAP ASE data type in template table
blob	image
date	datetime
datetime	datetime
decimal	decimal
double	float
int	int
interval	real
long	text
numeric	decimal
real	real
time	datetime

Data Services data type	SAP ASE data type in template table
varchar	varchar
timestamp	datetime

7.2.2.19 SAP Sybase IQ

The following table shows the conversion between SAP Sybase IQ data types and Data Services data types when Data Services imports metadata from a Sybase IQ source or target into the repository and when it loads data into an external table or file.

SAP Sybase IQ data type	Converts to or from Data Services data type
bigint	decimal
binary	not supported
bit	int
blob	blob
char	varchar
clob	long
date	date
datetime	datetime
decimal	decimal
double	double
float	real
int	int
long binary	blob
long varchar	long
money	decimal(19,4)
numeric	decimal
real	real
rowid	decimal
smalldatetime	datetime
smallint	int
smallmoney	decimal(10,4)
time	time
timestamp	datetime
tinyint	int
unsigned bigint	double

SAP Sybase IQ data type	Converts to or from Data Services data type
unsigned int	int
varbinary	not supported
varchar	varchar

The following table shows the conversion from internal data types to SAP Sybase IQ data types in template tables or Data_Transfer transform tables.

Data Services data type	SAP Sybase IQ data type in template table
blob	longbinary
date	date
datetime	timestamp
decimal	decimal
double	double
int	int
interval	float
long	N/A
numeric	N/A
real	real
time	time
varchar	varchar
timestamp	timestamp

7.2.2.20 SAP Vora data type conversions

SAP Vora has different data types than SAP Data Services. Therefore, Data Services must perform data conversion upon reading data from and loading data to SAP Vora tables.

The following table shows the conversion between SAP Vora data types and Data Services data types.

SAP Vora data type to SAP Data Services data type

SAP Vora data type	SAP Data Services data type
integer	int
tinyint	int
smallint	int
bigint	decimal

SAP Vora data type	SAP Data Services data type
char	varchar
varchar	varchar
real	real
double	double
decimal	decimal
boolean	int

SAP Data Services data type to SAP Vora data type

SAP Data Services data type	SAP Vora data type
int	integer
varchar	varchar
interval	real
real	real
double	double
decimal	decimal
date	date
time	time
datetime	timestamp
timestamp	timestamp
blob	varchar
long	varchar

Related Information

[SAP Vora datastore \[page 121\]](#)

7.2.2.21 Teradata

The following table shows the conversion between Teradata data types and Data Services data types when Data Services imports metadata from a Teradata source or target into the repository and when it loads data into an external table or file.

Teradata data types	Converts to or from Data Services data type
bigint	decimal
blob	blob
byteint	int
clob	long
char varying (n)	varchar(n)
char [(n)]	varchar(n)
date	date
decimal	decimal
double precision	float
float	float
int	int
long varchar	long
long vargraphic	long
numeric	decimal
real	float
smallint	int
time	time
timestamp	datetime
varchar < 32000	varchar
varchar >= 32000	long
varbyte < 32000	not supported
varbyte >= 32000	blob

The following table shows the conversion from internal data types to Teradata data types in template tables or Data_Transfer transform tables.

Data Services data type	Teradata data type in template table
blob	blob
date	date
datetime	timestamp
decimal	decimal

Data Services data type	Teradata data type in template table
double	N/A
int	int
interval	N/A
long	long varchar
numeric	N/A
real	N/A
time	time
varchar	varchar
timestamp	N/A

7.2.3 Conversion of data types within expressions

When possible, SAP Data Services optimizes data flows by pushing expressions down to an underlying database manager. In a single transaction, the software can push down expressions so that they are performed by the underlying database manager. However, when the software evaluates an expression which includes operands of more than one data type, the software attempts to convert the operands to the same data type first. (Except for national character-set data types which can be pushed down while others in an expression are not. For more information about supported national character-set data types, see [varchar \[page 316\]](#)).

When a conversion is required, the software provides a message at validation.

If the conversion is illegal, the software provides an error and you must remove the mismatch before executing the job.

If the conversion is legal, the software provides a warning indicating that it will not interrupt job execution.

i Note

When the software converts a data type to evaluate an expression, the results might not be what you expect. To avoid legal but incorrect conversions, always validate before executing and examine the circumstance of any data type conversion warnings.

7.2.4 Conversion among number data types

SAP Data Services uses a type-promotion algorithm to evaluate expressions that contain more than one number data type. Number data types are ranked from highest to lowest, as follows:

- decimal
- double
- real
- int

If the software encounters expressions that have more than one number data type among the operands, it converts all of the operands to the data type of the operand with the highest ranking type.

For example, if A is an `int` and B is a `double`, the expression `A+B` is evaluated by first converting A to `double` and then adding the two `double` values. The result is type `double`.

If in addition to A and B, you multiply the result by a `decimal` number C, then `(A+B)*C` is evaluated by first converting `(A+B)` to `decimal`, and then performing the indicated operations on the two `decimal` values. The result is type `decimal`.

For addition, subtraction, and multiplication, the operation result will be equal to the higher of the two operands. For example:

```
int + double = double
```

The following algorithm is used for division:

	Numerator data type				Denominator data type			
	int	real	double	decimal(p,s1)	int	real	double	decimal(p,s1)
int	double	double	double	decimal(p,s1)	double	double	double	decimal(p,s1)
real	double	double	double	decimal(p,s1)	double	double	double	decimal(p,s1)
double	double	double	double	decimal(p,s1)	double	double	double	decimal(p,s1)
decimal(p,s2)	decimal(p,s2)	decimal(p,s2)	decimal(p,s2)	decimal(p,s2)	decimal(p,s2)	decimal(p,s2)	decimal(p,s2)	decimal(p,max(10,s1,s2))

Conversion among decimals of different scale or precision

If decimals of two different scales are included in a single expression, the software uses the higher of the two scales. For example:

```
decimal(5,4) * decimal(7,2) = decimal(7,2)
```

Expect a loss of precision when operating on two `decimals` of different scale values. For example, when adding a `decimal(28, 26)` to a `decimal(28, 1)`, the resulting decimal value has the lower of the two scale values:

```
400000.5 + 40.00005 = 400040.5
```

The least scale for division involving a decimal is 10.

Conversions between strings and numbers

When the software encounters a string where a number would normally be expected (for example, in mathematical operations or functions that expect numeric arguments), it will attempt to convert the string to a number.

For multiplication and division operations, operands are converted to numbers. Other promotion algorithms are shown in the following table.

Addition

Provided data type	Data type required to evaluate the expression		
	Number	Date/time	String
Number	OK (promoted)	Number to interval	String to number
Date/time	Number to interval	Illegal	String to interval
String	String to number	String to interval	String to real

Subtraction

Provided data type	Data type required to evaluate the expression			
	Number	Date/time	String	Interval
Number	OK (promoted)	Illegal	String to number	Interval to number
Date/time	Number to interval	OK	String to interval	OK
String	String to number	String to datetime	String to real	String to number
Interval	Number to interval	Illegal	String to interval	OK

Comparison

Provided data type	Data type required to evaluate the expression			
	Number	Date/time	String	Interval
Number	OK (promoted)	Illegal	String to number	OK
Date/time	Illegal	OK	Illegal	Illegal
String	Illegal	String to datetime	OK	String to interval
Interval	OK	Illegal	String to interval	OK

Conversions between strings and dates

For Oracle, if you load datetime data from SAP Data Services into a char field in an Oracle table, Oracle puts the data in its default datetime format—which includes only date values—and loses the time from the value.

7.2.5 Conversion between explicit data types

You can use functions to convert data from one type to another:

- [interval_to_char](#) [page 1114]
- [julian_to_date](#) [page 1128]
- [num_to_interval](#) [page 1176]
- [to_char](#) [page 1221]

- [to_date](#) [page 1224]
- [to_decimal](#) [page 1225]

You can also import database-specific functions to perform data type conversions.

7.2.6 Conversion between native data types

A Data Quality transform can get and set field data in a format other than the declared data type. For example, if the input field is varchar, it can be mapped to an int data type field, as long as the varchar field contains all digits. However, certain conversions are not supported depending on the data type and field content.

Note

When a data type is mapped to an input or output field that is an invalid data type, the transform issues a verification error.

Example

The USA Regulatory Address Cleanse transform has a varchar type input field named Postcode_Full. Varchar field types can write to any kind of data type as long as the data is formatted correctly, and contains all digits. The Postcode_Full field could be integer because the field contains numbers. However, the Postcode_Full field could not be date type because it does not conform to the date format.

The remaining portion of this section lists each field type and the applicable and invalid data types for each.

7.2.6.1 date

Input: A transform can read from a date input field to the following data types:

- date
- character
- datetime

Output: A transform may write to the following data types from a date data type:

- date
- character
- datetime

Invalid input and output data types for date:

- integer
- double
- decimal

7.2.6.2 datetime

Input: A transform can read from a datetime input field to the following data types:

- datetime
- character
- date (with possible truncation)
- time (with possible truncation)

Output: A transform may write to the following data types from a datetime data type:

- datetime
- character
- date (with possible truncation)
- time (with possible truncation)

Invalid input and output data types for datetime:

- integer
- double
- decimal

7.2.6.3 decimal

Input: A transform can read from a decimal input field to the following data types:

- decimal
- character
- integer (data may be truncated)
- double (data may be truncated)

Output: A transform may write to the following data types from a decimal data type:

- decimal
- character
- integer (with possible truncation)
- double (with possible truncation)

Invalid input and output data types for decimal:

- date
- time
- datetime

7.2.6.4 double

Input: A transform can read from a double input field and write to the following data types:

- double
- character
- decimal (with possible truncation)
- integer (with possible truncation)

Output: A transform may write to the following data types from a double data type:

- double
- character
- decimal (with possible truncation)
- integer (with possible truncation)

Invalid input and output data types for double:

- date
- time
- datetime

7.2.6.5 int (integer)

Input: A transform can read from an int (integer) input field to the following data types:

- integer
- character
- decimal
- double

Output: A transform may write to the following data types from an int data type:

- integer
- character
- double
- decimal

Invalid input and output data types for int:

- date
- time
- datetime

7.2.6.6 varchar

Input: A transform may read input varchar data to any other supported data type. However, if the varchar data is not formatted correctly for the data type, the results are undefined. For example, if a varchar data type is converted to integer, it must contain all digits to convert correctly.

Output: A transform may write any supported data type to a varchar data type. The transform automatically converts the field contents to varchar data.

i Note

The data may be truncated if the output field is not long enough.

8 Transforms

A transform is a step in a data flow that acts on a data set. Built-in Data Services transforms are available through the object library in Designer.

The transforms described in this section are available from the Local Object Library in Designer in the [Transforms](#) tab.

The transforms that you have available in the [Transforms](#) tab depend on your Data Services package. If a transform belongs to a package that you have not purchased, it is unavailable and cannot be used in a Data Services job.

Transforms are grouped into the following categories:

- Data Integrator
- Data Quality
- Platform
- Text Data Processing

Related Information

[Transform reference \[page 351\]](#)

[Dynamic transform settings \[page 354\]](#)

[Data Services Embedded Help \[page 356\]](#)

8.1 Transform reference

SAP Data Services has several groups of transform types, and each group contains individual transforms that provide you with the ability to transform your data in many ways.

The following tables list the transforms in each group with a brief description.

Data Integrator transforms

Transform	Description
Data_Transfer	Allows a data flow to split its processing into two sub data flows and push down resource-consuming operations to the database server.
Date_Generation	Generates a column filled with date values based on the start and end dates and increment that you provide.
Effective_Date	Generates an additional "effective to" column based on the primary key's "effective date."

Transform	Description
Hierarchy_Flattening	Flattens hierarchical data into relational tables so that it can participate in a star schema. Hierarchy flattening can be both vertical and horizontal.
History_Preserving	Converts rows flagged as UPDATE to UPDATE plus INSERT, so that the original values are preserved in the target. You specify in which column to look for updated data.
Key_Generation	Generates new keys for source data, starting from a value based on existing keys in the table you specify.
Map_CDC_Operation	Sorts input data, maps output data, and resolves before- and after-images for UPDATE rows. While commonly used to support Oracle changed-data capture, this transform supports any data stream if its input requirements are met.
Pivot (Columns to Rows)	Rotates the values in specified columns to rows. (Also see Reverse Pivot.)
Reverse Pivot (Rows to Columns)	Rotates the values in specified rows to columns.
Table_Comparison	Compares two data sets and produces the difference between them as a data set with rows flagged as INSERT and UPDATE.
XML_Pipeline	Processes large XML inputs in small batches.

Data Quality transforms

Transform	Description
Address Lookup	Completes and populates addresses with minimal data, and can offer suggestions for possible matches.
Associate	Compares group numbers to find associated matches from different Match transforms.
Country_ID	Parses input data and then identifies the country of destination for each record.
Data_Cleanse	Identifies and parses name, title, and firm data, phone numbers, Social Security numbers, dates, and e-mail addresses. It can assign gender, add prenames, generate Match standards, and convert input sources to a standard format. It can also parse and manipulate various forms of international data, as well as operational and product data.
DSF2 Walk Sequencer	Adds delivery sequence information to your data, which you can use with presorting software to qualify for walk-sequence discounts.
Geocoder	Identifies and appends geographic information to address data such as latitude and longitude.
Global_Address_Cleanse	Identifies, parses, validates, and corrects global address data, such as primary number, primary name, primary type, directional, secondary identifier, and secondary number.
Match	Compares records, based on your criteria, or business rules, to find matching records in your data.
USA_Regulatory_Address_Cleanse	Identifies, parses, validates, and corrects USA address data according to the U.S. Coding Accuracy Support System (CASS).

Platform transforms



Transform	Description
Case	Simplifies branch logic in data flows by consolidating case or decision making logic in one transform. Paths are defined in an expression table.

Transform	Description
Data_Mask	<p>Uses data masking techniques to disguise or hide personal information contained in your databases. For example, bank account numbers, credit card numbers, and income. Data masking techniques include the following:</p> <ul style="list-style-type: none"> • Character replacement • Number variance • Date variance • Pattern variance • Number generalization • Date generalization <p>Data masking maintains data relevancy and relationships while keeping client information confidential and anonymous, and helps support your business data protection policies.</p>
Map_Operation	Modifies data based on current operation codes and mapping expressions. The operation codes can then be converted between data manipulation operations.
DQM_Microservices	Configures and executes a subset of DQM microservices services within Data Services.
Merge	Unifies rows from two or more sources into a single target.
Query	Retrieves a data set that satisfies conditions that you specify. A Query transform is similar to a SQL SELECT statement.
Row_Generation	Generates a column filled with integer values starting at zero and incrementing by one to the end value you specify.
SQL	Performs the indicated SQL query operation.
User_Defined	Does just about anything that you can write Python code to do. You can use the User-Defined transform to create new records and data sets, or populate a field with a specific value, just to name a few possibilities.
Validation	Ensures that the data at any stage in the data flow meets your criteria. You can filter out or replace data that fails your criteria.

Text Data Processing transforms

Transform	Description
Entity_Extraction	Extracts information (entities and facts) from unstructured data and creates structured data that can be used by various business intelligence tools.

i Note

For all transforms, to refresh a target schema after making changes to transform options, choose [View](#)  [Refresh](#)  or press **F5**.

Related Information

[Data Integrator transforms \[page 357\]](#)

[Data Quality transforms \[page 437\]](#)

[Platform transforms \[page 833\]](#)

[Text Data Processing transforms \[page 1013\]](#)

8.2 Dynamic transform settings

Dynamic transform settings allow you to change a transform's settings after the transform is initialized, without having to terminate and reinitialize the transform. You can pass each new setting through an input field to the transform. The transform will get an updated setting from the input field and adjust its processing to use the new setting, before processing the incoming record.

The settings a transform is initialized with are considered the transform's default settings. Dynamic setting values that are specified in the input fields are only valid for that record and do not affect any subsequent record. If the value specified for a given option is NULL or blank, then the record will be processed with the default setting for that option. If the dynamic setting is invalid, then the transform will log a warning and then use the default settings.

The dynamic input fields in Data Services are:

Transform	Dynamic input field
Data Cleanse	<p>Option_Content_Domain_Sequence</p> <p>The domain must be specified as an abbreviation of the domain. The valid predefined values are: AR, ZH, CS, DA, NL, EN_US, EN_GB, EN_AU, EN_IN, FR, DE, HU, ID, IT, JA, MS, NO, PL, PT_BR, PT_PT, RO, RU, SK, ES_MX, ES_ES, SV, TR, and GLOBAL.</p> <p>The Global domain is a special content domain which contains all variations and their associated properties. If a variation is not associated with domain-specific information the Global domain serves as the default domain. The Global domain is required for every content domain sequence. Be sure to add GLOBAL as the last domain in the sequence.</p> <p>The Content domain sequence input field may hold more than one domain. If there is more than one domain, you must separate the domains with a pipe (). For example, to specify the domain for Spain and then for Portugal, enter ES_ES PT_PT GLOBAL.</p>
Data Cleanse	<p>Option_Country</p> <p>Contains an ISO2 country code to help parse phone data and to determine the content domain sequence. You can obtain ISO2 country code data either from an input file that contains the codes, or from an upstream transform such as Global Address Cleanse that is set up to output ISO2 codes.</p> <ul style="list-style-type: none">• Parse phone data: Data Cleanse uses the ISO2 country code to help determine the country code for parsing phone data when the Option_Country dynamic input field is populated with an ISO2 country code, and the country is included in the cleansing package. Map the Global Address Cleanse field ISO_Country_Code_2Char to the Option_Country input field.• Set domain sequence: Data Cleanse uses the ISO2 code to determine content domain sequence when the Option_Country dynamic input field is populated with an ISO2 code. The transform automatically generates the output format based on the ISO2 country code and values in the Option_Language and Option_Region fields.

Transform	Dynamic input field
Data Cleanse	<p>Option_Language Option_Region</p> <p>Use these two fields with the Option_Country field to help determine the content domain and output format.</p> <p>There are a few countries where Option_Language and Option_Region data is helpful to determine the most appropriate content domain and output format. The two fields are applicable only to certain countries. For example, Switzerland, Belgium, or Canada can be used for Option_Language and Option_Region.</p> <p>The transform uses these fields only to determine the most appropriate content domain and output format</p> <p>The Option_Country, Option_Language, and Option_Region input fields should be mapped from the following Global Address Cleanse output fields in this order:</p> <ul style="list-style-type: none"> • ISO_Country_Code_2Char • Language • Region1 <p>These three dynamic Data Cleanse input fields work together to determine the Option_Content_Domain_Sequence and/or Option_Output_Format. For more information and examples, see “About Domains” in the <i>Designer Guide</i>.</p>
Data Cleanse	<p>Option_Output_Format</p> <p>The output format must be specified as an abbreviation of the domain. The valid predefined values are: AR, ZH, CS, DA, NL, EN_US, EN_GB, EN_AU, EN_IN, FR, DE, HU, ID, IT, JA, MS, NO, PL, PT_BR, PT_PT, RO, RU, SK, ES_MX, ES_ES, SV, and TR.</p>
Entity Extraction	Language
Geocoder	Option_Distance_Unit
Geocoder	Option_Max_Records
Geocoder	Option_Radius
Global Address Cleanse	Option_Canada_Output_Address_Language
Global Address Cleanse	Option_GAC_Dual_Address
Global Address Cleanse	Option_Standardization_Address_Line_Alias
Global Address Cleanse	Option_Standardization_Assign_Locality
Global Address Cleanse	Option_Standardization_Capitalization
Global Address Cleanse	Option_Standardization_Character_Width_Style
Global Address Cleanse	Option_Standardization_Directional_Style
Global Address Cleanse	Option_Standardization_Include_Locality_Addition
Global Address Cleanse	Option_Standardization_Locality_Name_Style
Global Address Cleanse	Option_Standardization_Output_Country_Language
Global Address Cleanse	Option_Standardization_Postal_Phrase_Style

Transform	Dynamic input field
Global Address Cleanse	Option_Standardization_Primary_Type_Style
Global Address Cleanse	Option_Standardization_Region_Style
Global Address Cleanse	Option_Standardization_Secondary_Description_Style
Global Address Cleanse	Option_Standardization_Secondary_Number_Style
Match	Option_Field_Algorithm_Geo_Proximity_<logical_name>_Max_Distance

8.3 Data Services Embedded Help

Embedded help is available for certain transforms in Designer.

The embedded help is the place to look when you need more information about Data Services transforms and options. The topic changes to help you with the context you're currently in. When you select a new transform or a new option group, the topic updates to reflect that selection.



You can also navigate to other topics by using hyperlinks within the open topic, or by using the table of contents.

The following transforms contain embedded help:

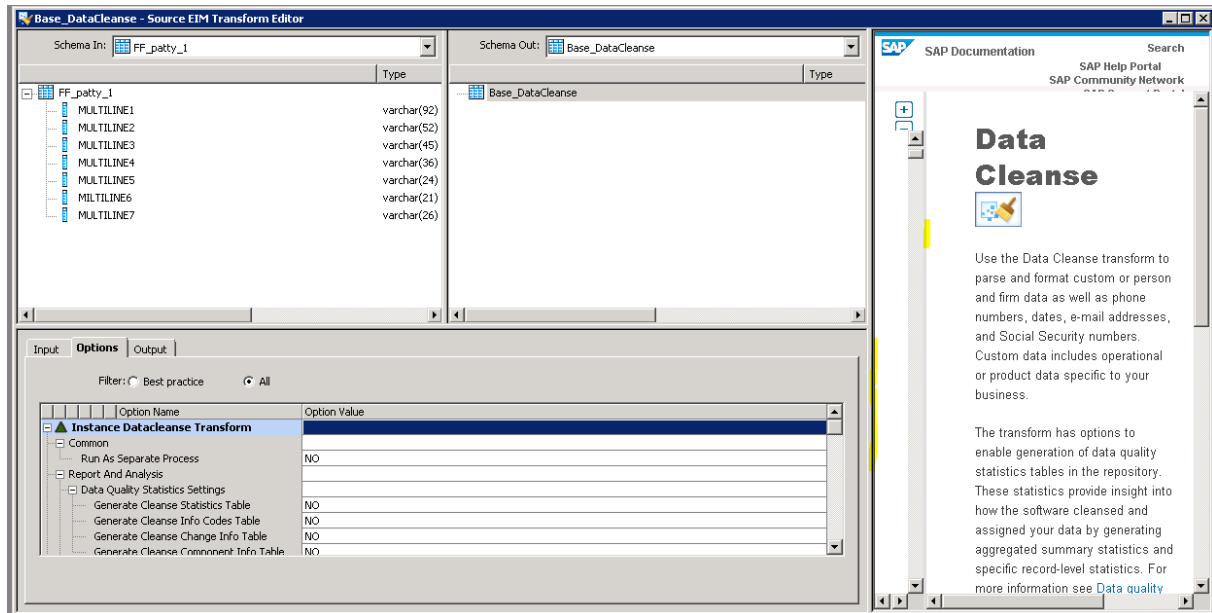
- Country ID
- Data Cleanse
- DQM Microservices
- Data Mask
- DSF2 Walk Sequencer
- Extraction
- Geocoder
- Global Address Cleanse
- Global Suggestion List
- Match Wizard
- USA Regulatory Address Cleanse
- Common

Note

To view option information for the Associate, Match, and User Defined transforms:

1. Select the transform in a data flow.
2. Select  **Tools**  **<transform> Editor** .

If the Help pane does not appear to the right of the transform editor pane, drag the right border of the editor to the left. (Shown highlighted in the following screen capture). To view or reduce the help table of contents, drag the border between the help pane and table of contents pane (also shown highlighted).



8.4 Data Integrator transforms

Transforms that allow you to extract, transform, and load data.

These transforms help ensure data integrity and maximize developer productivity for loading and updating data warehouse environment.

[Data_Transfer \[page 358\]](#)

Use the Data_Transfer transform in a data flow to push down certain operations to the database server for more efficient processing.

[Date_Generation \[page 372\]](#)

Use the Date_Generation transform in a data flow to produce a series of dates incremented as you specify.

[Effective_Date \[page 375\]](#)

Use the Effective_Date transform to calculate an “effective-to” value for data that contains an effective date.

[Hierarchy_Flattening \[page 380\]](#)

Use the Hierarchy_Flattening transform to take input with parent child relationships and output details about the hierarchy.

[History_Preserving \[page 392\]](#)

Use the History_Preserving transform to flag rows so that the software preserves the original values in the target.

[Key_Generation \[page 401\]](#)

Use the Key_Generation transform to generate new keys for source data that starts from a value based on an existing key in the table.

[Map_CDC_Operation \[page 405\]](#)

Use the Map_CDC_Operation transform to sort input data, map output data, and resolve before- and after-images for UPDATE rows.

[Pivot \(Columns to Rows\) \[page 413\]](#)

Use the Pivot transform to rotate the values in specified columns to rows.

[Reverse Pivot \(Rows to Columns\) \[page 418\]](#)

Use the Reverse Pivot to rotate the values in specified rows to columns.

[Table_Comparison \[page 422\]](#)

Use the Table_Comparison transform to compare two data sets and produce the difference between them as a data set with rows flagged as INSERT and UPDATE.


[XML_Pipeline \[page 433\]](#)

Use the XML_Pipeline transform to process large XML inputs in small batches.

8.4.1 Data_Transfer

Use the Data_Transfer transform in a data flow to push down certain operations to the database server for more efficient processing.

Data_Transfer information

Characteristic	Description
	Data_Transfer icon
Use	<p>Writes the data from a source or the output from another transform into a transfer object and then reads data from the transfer object. The transfer type can be a relational database table or file.</p> <p>Pushes down resource-consuming operations to the database server. Transfer type is a database table. Example resource-consuming operations are joins, GROUP BY, and sorts.</p>
Data inputs	<p>Input can be from a source or the output data set from another transform with rows flagged with the NORMAL operation code. This data is referred to as the input data set.</p> <p>The input data set cannot contain hierarchical (nested) data.</p>

Characteristic	Description
Data outputs	<p>Output can be a data set with the same schema and the same operation code as the input data set. If a subsequent ORDER BY or GROUP BY operation is pushed down to the database, the output rows are in the ORDER BY or GROUP BY order.</p> <p>Data Services automatically splits the data flow into secondary data flows and executes them serially. The secondary data flow names use the following format:</p> <pre><dataflowname_n></pre> <p>The variable <code><n></code> is the number of the secondary data flow.</p>

[Data_Transfer options \[page 359\]](#)

Use the Data_Transfer editor to specify the transfer type and options associated with that type.

[Target editor options \[page 360\]](#)

Tabs in the Target options appear based on the transfer type that you choose.

[Using Data_Transfer in a data flow \[page 370\]](#)

Increase the efficiency of a Group By operation in a dataflow by adding a Data_Transfer transform.

Parent topic: [Data Integrator transforms \[page 357\]](#)

Related Information

[Date_Generation \[page 372\]](#)

[Effective_Date \[page 375\]](#)

[Hierarchy_Flattening \[page 380\]](#)

[History_Preserving \[page 392\]](#)

[Key_Generation \[page 401\]](#)

[Map_CDC_Operation \[page 405\]](#)

[Pivot \(Columns to Rows\) \[page 413\]](#)

[Reverse Pivot \(Rows to Columns\) \[page 418\]](#)

[Table_Comparison \[page 422\]](#)

[XML_Pipeline \[page 433\]](#)

8.4.1.1 Data_Transfer options

Use the Data_Transfer editor to specify the transfer type and options associated with that type.

Depending on the transfer type you select, additional tabs appear.

When the *Transfer type* is Table and the *Database type* is any relational database management system, the software displays the following additional tabs:

- Options
- Bulk Loader Options
- Pre-Load Commands
- Post-Load Commands

The *Options* tab displays the data definition language (DDL) that the software uses to create the table. Modify the create table statement to add clauses such as EXTENTSIZE. Also, save the DDL to execute later.

Parent topic: [Data_Transfer \[page 358\]](#)

Related Information

[Target editor options \[page 360\]](#)

[Using Data_Transfer in a data flow \[page 370\]](#)

8.4.1.2 Target editor options

Tabs in the Target options appear based on the transfer type that you choose.

The tabs that appears on the target editor depend on the transfer type that you specify. Tabs in the target editor include:

- General
- Options
- Bulk Loader Options
- Pre-Load and Post-Load Commands

[General tab option descriptions \[page 361\]](#)

The options that you complete in the *General* tab in the target editor are based on the transfer type.

[Options tab descriptions \[page 365\]](#)

The options that you complete in the *Options* tab in the target editor are based on the table - relational database transfer type.

[Bulk Loader Options tab descriptions \[page 367\]](#)

Use the options in the *Bulk Loader Options* tab when bulk loading is applicable for the specific relational database management system.

[Pre-Load and Post-Load Commands tabs option descriptions \[page 368\]](#)

Use the *Pre-Load Commands* tab and the *Post-Load Commands* tab to specify SQL commands that Data Services executes before starting a load or after finishing a load into a transfer table.

[Editing SQL command lines \[page 369\]](#)

Edit the SQL commands when you work with SQL commands in the pre and postload command tabs.

Parent topic: [Data_Transfer \[page 358\]](#)

Related Information

[Data_Transfer options \[page 359\]](#)

[Using Data_Transfer in a data flow \[page 370\]](#)

8.4.1.2.1 General tab option descriptions

The options that you complete in the *General* tab in the target editor are based on the transfer type.

Transfer types determine the options that appear in the *General* tab. Types include:

- File
- Table
- Automatic
- Table, relational

The following table indicates the transfer type for each option.

General tab option descriptions

Transfer type	Option	Description
File, Table, or Automatic	<i>Enable transfer</i>	Enables or disables the execution of the Data_Transfer transform. It is selected by default. Disable this transform to compare performance with transfer enabled and disabled. <div>i Note When you run the job in debug mode, Data Services automatically disables all Data_Transfer transforms.</div>
File	<i>File options: File name</i>	Name of the flat file that you want to use as transfer for sub data flows. The file does not need to exist.
File	<i>File options: Root directory</i>	The name of the root directory that contains the file to use for transfer. If your default Job Server and Designer reside on the same computer, you can use the browse button to find the <i>Root directory</i> . If your default Job Server does not reside on your local computer, enter the path to your <i>Root directory</i> manually. You can use a global variable or parameter for the pathname.
File, Table, or Automatic	<i>Join rank</i>	Indicates the rank of the output data set relative to other tables and files joined in a data flow. The software joins sources with higher join ranks before joining sources with lower join ranks. Join rank specified in the Query transform editor FROM tab overrides any join rank specified in a source. For new jobs, specify the join rank only in the Query transform editor. Must be a non-negative integer. Default value is 0.

Transfer type	Option	Description
Table - relational	Table options: Array fetch size	<p>Indicates the number of rows retrieved in a single request to a source database. The default value is 1000. Higher numbers reduce requests, lowering network traffic, and possibly improve performance. The maximum value is 5000.</p> <p>This option is available for source tables from DB2, Informix, ODBC, Oracle, and SQL Server datastores.</p> <p>When retrieving a column with an Oracle long data type, Data Services automatically sets Array Fetch Size to 1. If a column has an Oracle long data type, Data Services can only retrieve one row at a time.</p>

Transfer type	Option	Description
Table - relational	Table options: Database type	<p>Determines the content for additional tabs on the Data_Transfer transform editor. Allows you to quickly set target option values in data flows.</p> <p>If your target datastore has multiple configurations, the target editor lists database types that you define for your datastore configuration. To add or remove items in this list, edit the datastore configuration information in the datastore editor.</p> <p>Data Services allows you to use target table editor option values from any datastore configuration based on the following rules:</p> <ul style="list-style-type: none"> • Datastore has one configuration: The software sets target table editor values to the default values for the configuration. • Datastore has more than one configuration with different database types or versions: The software determines the initial values for the additional database types or versions from the Use values from option in Create New Configuration dialog box. The Create New Configuration dialog box is in the datastore editor. If you select Restore values if they already exist in Create New Configuration, the software uses values that were previously defined for the database type or version. <div> <p>i Note</p> <p>It is possible for a data flow to contain target table editor values for a database type or version, even if its datastore configuration was deleted. Data Services retains all target table editor values saved with every datastore configuration. If such values exist, then it restores those values. Otherwise, it gets the values from the configuration you select from the Use values from option.</p> </div> <div> <p>❖ Example</p> <p>Suppose that you set a configuration for Oracle 8i. When you edit the target table editor options, you change the Rows Per Commit default value of 1000 to 500. Later you add a Microsoft SQL Server 2000 database datastore configuration to your original datastore and set the Use values from option to Oracle 8i. The target table editor settings for SQL Server inherit the value 500 for Rows per Commit because 500 was the value set in the Oracle 8i configuration.</p> </div>

Transfer type	Option	Description
Table - relational	<i>Table options: Table name</i>	<p>Name of the database table that you want to use as transfer for sub data flows. Specify the table name with the following format:</p> <pre><datastorename.ownername.tablename or <datastorename.ownername.schemaname.table name></pre> <p>You can click the browse button (ellipses) to display your datastores. Select a table name from the list or type in the name of a new table.</p>
File, Table, or Automatic	<i>Transfer type</i>	<p>Choose one of the following transfer types to temporarily store the data of each sub data flow:</p> <ul style="list-style-type: none"> <i>Table</i>: Database table from an existing datastore. Specify the Table options in <i>Table name</i>, <i>Database type</i>, and <i>Array fetch size</i>. <i>File</i>: A flat file. Specify the File options in <i>Root directory</i> and <i>File name</i>. <i>Automatic</i>: Data Services optimizer chooses the transfer type from either: <ul style="list-style-type: none"> Your datastores that have the <i>Enable automatic data transfer</i> checkbox selected. The pageable cache directory that you specify in the Server Manager. <p>The Data Services optimizer chooses the transfer type and location that could provide the optimal performance, based on subsequent operations that the data flow contains.</p> <div> <p>❖ Example</p> <p>If an ORDER BY follows the Data_Transfer transform in a data flow, the optimizer might pick the database datastore that contains the data so that the ORDER BY can be pushed down to the database.</p> <p>If the data flow does not contain an ORDER BY, GROUP BY, DISTINCT, join, or any expression that can be pushed down, the Optimizer chooses the pageable cache directory. If multiple files are available (one on each job server in a server group), the optimizer chooses the directory that is local to the data flow process.</p> </div>

Parent topic: [Target editor options \[page 360\]](#)

Related Information

[Options tab descriptions \[page 365\]](#)

[Bulk Loader Options tab descriptions \[page 367\]](#)

[Pre-Load and Post-Load Commands tabs option descriptions \[page 368\]](#)

[Editing SQL command lines \[page 369\]](#)

8.4.1.2.2 Options tab descriptions

The options that you complete in the *Options* tab in the target editor are based on the table - relational database transfer type.

Option tab descriptions

Option	Description
<i>Data definition language (DDL)</i>	Edit or save the SQL CREATE TABLE statement that Data Services generates. Optionally add extra parameters such as table space name or extent size, or enter your own DDL statement. Data Services saves the DDL and uses it at job execution time.
<i>Delete data before loading</i>	Deletes the existing data in the table before loading. <ul style="list-style-type: none">Selected: Deletes table data before loading new data. This is the default.Not selected: Appends data to existing data in the table.
<i>Drop and re-create before loading</i>	Drops the existing table and creates a new table with the same name before loading. <ul style="list-style-type: none">Selected: Drops the existing table and creates a new table with the same name before loading. This is the default.Not selected: Does not drop and re-create the table before loading. <div>i Note Unlike a template table, you can use bulk loading options for a transfer table even when the <i>Drop and re-create before loading</i> option is selected.</div>
<i>Enable partitioning</i>	Displays only if the transfer table is either physically partitioned or logically partitioned. Enables Data Services to use the partition information in this transfer table. <ul style="list-style-type: none">Selected: Data Services transfers data using the number of partitions in the table as the maximum number of parallel instances.Not selected: Disables this feature. If you select <i>Enable partitioning</i> , you cannot select <i>Number of Loaders</i> .
<i>Generate default DDL</i>	Click to display the SQL CREATE TABLE statement that Data Services generates.

Option	Description
<i>Number of loaders</i>	<p>Specify the number of loaders.</p> <ul style="list-style-type: none"> Loading with one loader is called “single loader loading.” Loading with more than one loader is called “parallel loading.” <p>The default number of loaders is 1.</p> <p>If you select <i>Number of Loaders</i>, you cannot select <i>Enable partitioning</i>.</p> <p>When parallel loading, each loader receives the number of rows indicated in the Rows per commit option, in turn, and applies the rows in parallel with the other loaders.</p> <div> <p>❖ Example</p> <p>For example, if you choose a Rows per commit of 1000 and set the number of loaders to 3, the first 1000 rows are sent to the first loader. The second 1000 rows are sent to the second loader, the third 1000 rows to the third loader, and the next 1000 rows back to the first loader.</p> </div>
<i>Rows per commit</i>	<p>Specifies the transaction size in number of rows.</p> <p>If set to 1000, Data Services sends a commit to the underlying database every 1000 rows.</p> <p>This option is not available for targets in real time jobs.</p>

Parent topic: [Target editor options \[page 360\]](#)

Related Information

[General tab option descriptions \[page 361\]](#)

[Bulk Loader Options tab descriptions \[page 367\]](#)

[Pre-Load and Post-Load Commands tabs option descriptions \[page 368\]](#)

[Editing SQL command lines \[page 369\]](#)

8.4.1.2.3 Bulk Loader Options tab descriptions

Use the options in the [Bulk Loader Options](#) tab when bulk loading is applicable for the specific relational database management system.

Bulk Loader Option description

Transfer type	Option	Description
Relational database table	Varies based on database type.	For information about specific database types, see Target tables [page 226] .

Parent topic: [Target editor options \[page 360\]](#)

Related Information

[General tab option descriptions \[page 361\]](#)

[Options tab descriptions \[page 365\]](#)

[Pre-Load and Post-Load Commands tabs option descriptions \[page 368\]](#)

[Editing SQL command lines \[page 369\]](#)

8.4.1.2.4 Pre-Load and Post-Load Commands tabs option descriptions

Use the [Pre-Load Commands](#) tab and the [Post-Load Commands](#) tab to specify SQL commands that Data Services executes before starting a load or after finishing a load into a transfer table.

Pre-load and post-load command option descriptions

Transfer type	Option	Description
Table - relational database	SQL Commands Value	<p>When a data flow is called, Data Services opens all objects in the data flow. For example, queries, transforms, sources, and targets. Next, Data Services runs the target preload script. As a result, Data Services executes any preload SQL commands before processing any transform.</p> <div>Note Because Data Services executes the SQL commands as a unit of transaction, do not include transaction commands in preload or post-load SQL statements.</div> <p>You cannot use preload and postload SQL commands in a real time job.</p> <p>To edit SQL command lines, select the line in SQL Commands to display the line in the Value box. Edit the line text in the Value box.</p>

Parent topic: [Target editor options \[page 360\]](#)

Related Information

[General tab option descriptions \[page 361\]](#)

[Options tab descriptions \[page 365\]](#)

[Bulk Loader Options tab descriptions \[page 367\]](#)

[Editing SQL command lines \[page 369\]](#)

8.4.1.2.5 Editing SQL command lines

Edit the SQL commands when you work with SQL commands in the pre and postload command tabs.

The following steps include adding a new line, deleting an existing line, and including variables and parameters in preload or postload SQL statements.

1. Add a new line:
 - a. Determine the position for the new line.
 - b. Select the existing line immediately before or after the new line position.
 - c. Right-click and select *Insert Before* or *Insert After* as applicable.
 - d. Type the SQL command in the Value text box.
2. Delete a line:
 - a. Select the line in the *SQL Commands* box
 - b. Right click, and choose *Delete*.
3. Include variables and parameters in preload or postload SQL statements:
 - a. Enter the variables and parameters in either brackets, braces, or quotes.

Data Services translates each statement differently, writing a statement that depends on the variable or parameter type.

❖ Example

Entered statement	Variable value	Written statement
[\$X]	5	5
[\$X]	John Smith	John Smith
{ \$X }	5	5
{ \$X }	John Smith	John Smith
'\$X'	5	5
'\$X'	John Smith	John Smith

Task overview: [Target editor options \[page 360\]](#)

Related Information

[General tab option descriptions \[page 361\]](#)

[Options tab descriptions \[page 365\]](#)

[Bulk Loader Options tab descriptions \[page 367\]](#)

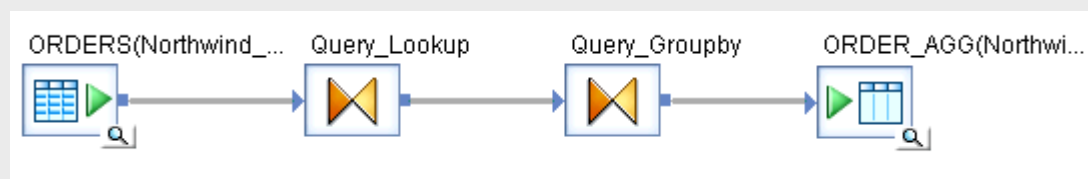
[Pre-Load and Post-Load Commands tabs option descriptions \[page 368\]](#)

8.4.1.3 Using Data_Transfer in a data flow

Increase the efficiency of a Group By operation in a dataflow by adding a Data_Transfer transform.

❖ Example

The following example shows a simple data flow that contains a Query transform. The Query transform looks up sales subtotals and groups the results by country and region.

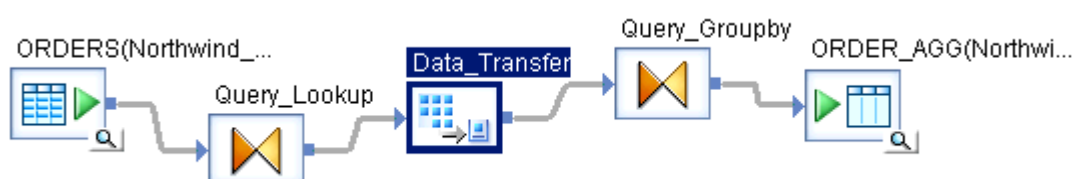


Suppose the GROUP BY operation processes millions of rows, which is slowing down processing. Data Services cannot push the GROUP BY operation down to the database because the Query transform that comes before the GROUP BY operation contains a lookup_ext function. Data Services cannot push down the lookup_ext function.

To make this data flow more efficient, add a Data_Transfer transform to split the lookup_ext function and the GROUP BY operation into two sub data flows. Then Data Services can push the GROUP BY operation to the target database.

The following steps show you how to restructure your data flow to accommodate the Data_Transfer transform:

1. Delete the connecting line between the two Query transforms.
2. Drag the Data_Transfer transform from the object library to the data flow in the work space and place it between the two Query transforms. Reconnect the objects.



3. Open the Data_Transfer transform editor and select the transfer type.
For this example, we select Table for *Transfer type*.
4. In the Table options area, click the browse button for *Table name*. Select the datastore that you want to use to contain the data to transfer to the second sub data flow.
5. Specify the transfer table in the Input table for Data_Transfer window.
 - a. Enter the name of the table that will contain the data to transfer. The table does not need to exist.
 - b. In the *Owner name* text box, type either the owner name or an alias.
If you specify an alias for the owner, the *CREATE TABLE* statement in the *Options* tab shows the alias name in square brackets. When you execute the job, Data Services replaces the alias name with the owner name.

- c. Click [OK](#).
6. Accept the default setting for [Array fetch size](#).
7. Save and execute the job.

When you execute the job, Data Services displays messages for each sub data flow. For the sample GROUP BY data flow, the following messages display for the data flow and sub data flows:

```
Job <LookupGroup_Orders_Job> is started.
Process to execute data flow<LookupGroup_Orders_DF> is started.
Starting sub data flow <LookupGroup_Orders_DF_1> on job server host <SJ-Comput>,
port <3508>. Distribution level <Job>.
Process to execute sub data flow <LookupGroup_Orders_DF_1> is started.
Sub data flow <LookupGroup_Orders_DF_1> is started.
Cache statistics for sub data flow <<LookupGroup_Orders_DF_1> are not available to
be used for optimization and need to be collected before they can be used.
Sub data flow <LookupGroup_Orders_DF_1> using PAGEABLE Cache with <1280 MB> buffer
pool.
Sub data flow <LookupGroup_Orders_DF_1> is completed successfully.
Process to execute sub data flow <LookupGroup_Orders_DF_1> is completed.
Starting sub data flow <LookupGroup_Orders_DF_2> on job server host <SJ-Comput>,
port <3508>. Distribution level <Job>.
Process to execute sub data flow <LookupGroup_Orders_DF_2> is started.
Sub data flow <LookupGroup_Orders_DF_2> is started.
Cache statistics determined that sub data flow <LookupGroup_Orders_DF_2> uses <0>
caches with a total size of <0> bytes. This is less than (or equal to) the virtual
memory <1610612736> bytes available for caches. Statistics is switching the cache
type to IN MEMORY.
Sub data flow <LookupGroup_Orders_DF_2> using IN MEMORY Cache.
Sub data flow <LookupGroup_Orders_DF_2> is completed successfully.
Process to execute sub data flow <LookupGroup_Orders_DF_2> is completed.
Process to execute data flow <LookupGroup_Orders_DF> is completed.
Job <LookupGroup_Orders_Job> is completed successfully.
```

Task overview: [Data_Transfer \[page 358\]](#)

Related Information


[Data_Transfer options \[page 359\]](#)

[Target editor options \[page 360\]](#)

8.4.2 Date_Generation

Use the Date_Generation transform in a data flow to produce a series of dates incremented as you specify.

Date_Generation information

Characteristic	Description
	Date_Generation icon
Use	Produces the key values for a time dimension target. From this generated sequence, populate other fields in the time dimension, such as day_of_week, using functions in a query.
Data inputs	N/A
Data outputs	<p>A data set with a single column named <code>DI_GENERATED_DATE</code> that contains the date sequence. The software flags the generated rows as INSERT.</p> <p>The Date_Generation transform does not generate hierarchical data.</p> <p>Generated dates can range from 1900.01.01 through 9999.12.31.</p>

[Date Generation option descriptions \[page 373\]](#)

Set specifications for the Date_Generation transform to use when it generates key values for a time dimension target.

[Example: Create time dimension target \[page 374\]](#)

An example showing how to use the Date_Generation transform to create a time dimension target.

Parent topic: [Data Integrator transforms \[page 357\]](#)

Related Information

[Data_Transfer \[page 358\]](#)

[Effective_Date \[page 375\]](#)

[Hierarchy_Flattening \[page 380\]](#)

[History_Preserving \[page 392\]](#)

[Key_Generation \[page 401\]](#)

[Map_CDC_Operation \[page 405\]](#)

[Pivot \(Columns to Rows\) \[page 413\]](#)

[Reverse Pivot \(Rows to Columns\) \[page 418\]](#)

[Table_Comparison \[page 422\]](#)

[XML_Pipeline \[page 433\]](#)

8.4.2.1 Date Generation option descriptions

Set specifications for the Date_Generation transform to use when it generates key values for a time dimension target.

Date_Generation option descriptions

Option	Description
<i>Cache</i>	<p>Indicates whether the software should read the required data from the source and load it into memory or pageable cache. Because an inner source of a join must be read for each row of an outer source, you might want to cache a source when it is used as an inner source in a join.</p> <ul style="list-style-type: none">• Yes: The source is always cached unless it is the outer-most source in a join. This is the default setting.• No: The source is never cached. <p>Cache specified in the Query transform editor <i>FROM</i> tab overrides any cache specified in a source. For new jobs, specify the cache in the Query transform editor only.</p>
<i>End date</i>	<p>The last date in the output sequence. Use the same format as the <i>Start date</i> to specify this date.</p>
<i>Increment</i>	<p>The interval between dates in the output sequence. Select <i>Daily</i>, <i>Monthly</i>, or <i>Weekly</i>.</p>
<i>Join rank</i>	<p>Indicates the rank of the output data set relative to other tables and files joined in a data flow. Must be a non-negative integer. The default is 0.</p> <p>The software joins sources with higher join ranks before joining sources with lower join ranks.</p> <p>Join rank specified in the Query transform editor <i>FROM</i> tab overrides any join rank specified in a source. For new jobs, specify the join rank in the Query transform editor only.</p>
<i>Start date</i>	<p>The first date in the output sequence. Specify this date using the following format: YYYY.MM.DD</p> <p>YYYY is a year value, MM is a month value, and DD is a day value.</p>

Parent topic: [Date_Generation \[page 372\]](#)

Related Information

[Example: Create time dimension target \[page 374\]](#)

8.4.2.2 Example: Create time dimension target

An example showing how to use the Date_Generation transform to create a time dimension target.

❖ Example

Follow these steps to create a time dimension target with dates from the beginning of the year 1997 to the end of the year 2000:

1. Place a Date_Generation transform, a query, and a target in a data flow.
2. Connect the output of the Date_Generation transform to the query, and the output of the query to the target.
3. Specify the following values in the Date_Generation transform:
 - **Start date:** 1997.01.01 (A variable can also be used.)
 - **End date:** 2000.12.31 (A variable can also be used.)
 - **Increment:** Daily (A variable can also be used.)

You can also specify a variable for these options.

4. Inside the query, create two target columns and the field name, and define a mapping for these time dimension values:
 - Business quarter: `BusQuarter`
Function: `quarter(Generated_date)`
 - Date number from start: `DateNum`
Function: `julian(Generated_date) - julian(1997.01.01)`

Parent topic: [Date_Generation \[page 372\]](#)


Related Information

[Date Generation option descriptions \[page 373\]](#)

8.4.3 Effective_Date

Use the Effective_Date transform to calculate an “effective-to” value for data that contains an effective date.

Effective_Date information

Characteristic	Description
	Effective_Date icon
Use	Calculates an effective-to value for data that contains an effective date. The calculated effective-to date and an existing effective date produce a date range that allows queries. The queries based on the effective dates produce meaningful results.
Data inputs	Data that has an effective date column. Effective dates allow you to indicate changes to information over time.
Data outputs	Includes all of the columns from the source schema and the calculated effective-to date column.

[Effective_Date transform editor \[page 376\]](#)

When you include the Effective_Date transform in a data flow, complete the applicable options in the editor.

[Effective_Date options \[page 376\]](#)

The Effective_Date transform editor has a tab in which you set effective date options.

[Example: Data inputs \[page 377\]](#)

Input data for an Effective_Date transform has an effective date column.

[Example: Data outputs \[page 379\]](#)

The Effective_Date transform output includes all of the columns from the source schema and the calculated effective-to date column.

Parent topic: [Data Integrator transforms \[page 357\]](#)

Related Information

[Data_Transfer \[page 358\]](#)

[Date_Generation \[page 372\]](#)

[Hierarchy_Flattening \[page 380\]](#)

[History_Preserving \[page 392\]](#)



[Key_Generation \[page 401\]](#)
[Map_CDC_Operation \[page 405\]](#)
[Pivot \(Columns to Rows\) \[page 413\]](#)
[Reverse Pivot \(Rows to Columns\) \[page 418\]](#)
[Table_Comparison \[page 422\]](#)
[XML_Pipeline \[page 433\]](#)

8.4.3.1 Effective_Date transform editor

When you include the Effective_Date transform in a data flow, complete the applicable options in the editor.

The Effective_Date transform editor includes the following sections:

- A *Schema In* pane on the left that shows the source schema
- A *Schema Out* pane on the right that shows the target schema
- A *Effective Date* tab that shows the transform options. Drag column names from the source schema to fill in values for the *Effective Date column* and *Effective sequence column* options.

The software generates the target schema based on the values that you choose in the transform options. To refresh the target schema after you change the options, choose  *View*  *Refresh* or press *F5*.

Parent topic: [Effective_Date \[page 375\]](#)

Related Information

[Effective_Date options \[page 376\]](#)
[Example: Data inputs \[page 377\]](#)
[Example: Data outputs \[page 379\]](#)

8.4.3.2 Effective_Date options

The Effective_Date transform editor has a tab in which you set effective date options.

Effective date tab option descriptions

Option	Description
<i>Default effective to date value</i>	A date assigned as the effective-to date for those rows with the highest effective date among related rows. You can also specify a variable for this option.

Option	Description
<i>Effective date column</i>	<p>A column in the input data set with the data type of <code>date</code> that contains the effective date. The software enters the column name automatically when it finds a column named <code>EFFDT</code> in the source. The column appears in the output schema with the name <code>EFFDT</code>.</p> <p>This field is required.</p>
<i>Effective sequence column</i>	<p>A column in the input data set that indicates the order in time of related rows that have duplicate effective dates. If no related rows share effective dates, the sequence numbers are the same, for example, 0. If related rows share the same effective dates, the software increments the sequence numbers as it adds rows with conflicting effective dates. The transform returns only the row containing the maximum sequence number if there are related rows with the same effective date.</p> <p>This field is required only if the input data set allows duplicate effective dates.</p>
<i>Effective to column</i>	<p>The name of the output column added to the output schema that contains the effective-to date, with a <code>date</code> data type.</p> <p>The effective-to date for a row is equal to the effective date of the related row with the closest greater effective date. If no such row exists, the <i>Default effective to date</i> is used.</p>

Parent topic: [Effective_Date \[page 375\]](#)

Related Information

[Effective_Date transform editor \[page 376\]](#)

[Example: Data inputs \[page 377\]](#)

[Example: Data outputs \[page 379\]](#)

8.4.3.3 Example: Data inputs

Input data for an `Effective_Date` transform has an effective date column.

Effective dates allow you to indicate changes to information over time. The effective date value in each row of a data set indicates the date from which the data in the row is valid. As changes are made to the information, more rows are included to describe the information as it changes over time. The effective date distinguishes each row that describes the set of information.

Example

An input data set includes the following columns:

- Project: Name that identifies the information being described.
- Effective date: The date from which the data in the row is valid.
- Status: Data that changes over time.

Project	Effective date	Status
Cherry Lake	1999.06.22	Proposal
Cherry Lake	2003.01.12	Case
Hetch Hetchy Reservoir	1999.08.02	Proposal
Hetch Hetchy Reservoir	2003.05.06	Case

The transform description uses the term “related rows” to refer to a set of rows that describe the same information as it changes over time. There are two sets of related rows in the example, described by the values in the Project column.

In the following table, an additional Effective sequence column contains a value that helps distinguish between rows with duplicate effective dates. The Hetch Hetchy Reservoir project contains a row with an Effective sequence of 0 and a row with an Effective sequence of 1. The different Effective sequence values indicate that the two rows have the same Effective date of 2002.10.17.

Project	Effective date	Effective sequence	Status
Cherry Lake	1999.06.22	0	Proposal
Cherry Lake	2002.01.12	0	Case
Hetch Hetchy Reservoir	1999.08.02	0	Proposal
Hetch Hetchy Reservoir	2002.10.17	0 ^a	Proposal
Hetch Hetchy Reservoir	2002.10.17	1 ^a	Case
Hetch Hetchy Reservoir	2003.05.06	0	Case

^a The effective sequence distinguishes the project status for two related rows with the same Effective date.

In this example, only the row with the largest sequence number of 1 is effective-dated by this transform.

If you construct a query that selects the status of the “Hetch Hetchy Reservoir” project on “2002.12.31”, the query returns the status from the last row, which is “Case.” The last row has an effective date of 2003.05.06. The Effective date for this row indicates a date that falls after the stated date of 2002.12.31. The other Hetch Hetchy Reservoir rows contain an Effective date that falls before the stated date.

Considerations

The input data set can contain only rows flagged as NORMAL. The input data set can contain hierarchical data. However, the transform operates only on the rows at the top-level of the input data set. The transform passes nested data through to the output without change. Columns that contain nested schemas cannot be used as transform parameters.

Parent topic: [Effective_Date \[page 375\]](#)

Related Information

[Effective_Date transform editor \[page 376\]](#)

[Effective_Date options \[page 376\]](#)

[Example: Data outputs \[page 379\]](#)

8.4.3.4 Example: Data outputs

The Effective_Date transform output includes all of the columns from the source schema and the calculated effective-to date column.

❖ Example

Given a default “effective-to” date of January 1, 2999, the software transforms the input described in the data input section as follows:

Project	Effective date	Effective-to date	Status
Cherry Lake	1999.06.22	2003.01.12	Proposal
Cherry Lake	2003.01.12	2999.01.01 ^a	Case
Hetch Hetchy Reservoir	1999.08.02	2002.10.17	Proposal
Hetch Hetchy Reservoir	2002.10.17	2003.05.06	Case
Hetch Hetchy Reservoir	2003.05.06	2999.01.01 ^a	Case

^a The default effective-to date closes the effective date range.

When an Effective sequence column is necessary to produce a unique key for related rows that contain the same Effective date, the output from the Effective_Date transform includes a single row where the input had more than one row with the same Effective date. The row returned contains the largest Effective sequence number of 1:

Project	Effective date	Effective to date	Effective sequence	Status
Cherry Lake	1999.06.22	2003.01.12	0	Proposal
Cherry Lake	2003.01.12	2999.01.01	0	Case

Project	Effective date	Effective to date	Effective sequence	Status
Hetch Hetchy Reservoir	1999.08.02	2002.10.17	0	Proposal
Hetch Hetchy Reservoir	2002.10.17	2003.05.06	1 ^a	Case
Hetch Hetchy Reservoir	2003.05.06	2999.01.01	0	Case
^a The transform in this example omits data from the row with the same Effective date and with the Effective sequence = 0 from the output.				

After the software generates a range of effective dates for a set of data, use the effective to date value to filter appropriate records. For example, extract the subset of records valid as of today by selecting only those records whose Effective to date column is later than the current date and Effective date column is earlier than current date.

The transform passes through nested schemas in the input without change.

Parent topic: [Effective_Date \[page 375\]](#)

Related Information

[Effective_Date transform editor \[page 376\]](#)


[Effective_Date options \[page 376\]](#)

[Example: Data inputs \[page 377\]](#)

8.4.4 Hierarchy_Flattening

Use the Hierarchy_Flattening transform to take input with parent child relationships and output details about the hierarchy.

Hierarchy_Flattening information

Characteristic	Description
	Hierarchy_Flattening icon

Characteristic	Description
Use	Constructs a complete hierarchy from parent/child relationships, then produces a description of the hierarchy in vertically or horizontally flattened format.
Data inputs	Rows that describe individual parent-child relationships.
Data outputs	Columns that represent each level in the hierarchy, with the root listed in the first column and the outermost leaf listed in the last column.

[Hierarchy_Flattening transform editor \[page 382\]](#)

When you include the Hierarchy_Flattening transform in a data flow, you configure the options in the editor.

[Hierarchy_Flattening options \[page 383\]](#)

Specify options in the Hierarchy_Flattening section to describe input parent child structure and the flattening type to use.

[Error conditions \[page 386\]](#)

If the hierarchy represented by the input data set is cyclic, Data Services produces a run-time error.

[Data inputs \[page 387\]](#)

Data input for the Hierarchy_Flattening transform includes rows that describe individual parent to child relationships.

[Data outputs: Horizontal flattening \[page 388\]](#)

Use horizontal flattening so that each row of the output describes a single node in the hierarchy and the path to that node from the root.

[Data outputs: Vertical flattening \[page 390\]](#)

Use vertical flattening so that each row of the output describes a single node in the hierarchy and the path to that node from the root.

Parent topic: [Data Integrator transforms \[page 357\]](#)

Related Information

[Data_Transfer \[page 358\]](#)

[Date_Generation \[page 372\]](#)

[Effective_Date \[page 375\]](#)

[History_Preserving \[page 392\]](#)

[Key_Generation \[page 401\]](#)

[Map_CDC_Operation \[page 405\]](#)

[Pivot \(Columns to Rows\) \[page 413\]](#)

[Reverse Pivot \(Rows to Columns\) \[page 418\]](#)

[Table_Comparison \[page 422\]](#)

[XML_Pipeline \[page 433\]](#)



[Data inputs \[page 387\]](#)

[Data outputs: Horizontal flattening \[page 388\]](#)

8.4.4.1 Hierarchy_Flattening transform editor

When you include the Hierarchy_Flattening transform in a data flow, you configure the options in the editor.

The Hierarchy_Flattening editor includes the following sections:

- *Schema In* pane contains the source schema
- *Schema Out* pane contains the target schema
- Hierarchy_Flattening transform options
 - Drag column names from the source schema to fill in values for the Parent column, Child column, Parent attribute list, and Child attribute list options.
 - Data Services generates the target schema based on the values that you choose in the transform options.
 - To refresh the target schema after you change the options, choose  [View](#)  or press *F5*.

Parent topic: [Hierarchy_Flattening \[page 380\]](#)

Related Information

[Hierarchy_Flattening options \[page 383\]](#)

[Error conditions \[page 386\]](#)

[Data inputs \[page 387\]](#)

[Data outputs: Horizontal flattening \[page 388\]](#)

[Data outputs: Vertical flattening \[page 390\]](#)

8.4.4.2 Hierarchy_Flattening options

Specify options in the Hierarchy_Flattening section to describe input parent child structure and the flattening type to use.

Option	Description									
<i>Child attribute list</i>	<p>Identifies a column or columns that are associated with the child column. Drag columns from the source schema into the <i>Child attribute list</i>. The column name appears in the target schema with a prefix that identifies the column as a child attribute.</p> <p>Specify columns, including nested schemas, as the child attribute.</p> <div><p>❖ Example</p><p>The following table shows the result of adding a column named <code>POPULATION</code> to the child attribute list for vertical and horizontal flattening.</p><table><tr><th>Flattening mode</th><th>Source column</th><th>Target column</th></tr><tr><td>Vertical</td><td>POPULATION</td><td>C_POPULATION</td></tr><tr><td>Horizontal</td><td>POPULATION</td><td>C_L1_POPULATION C_L2_POPULATION There is one column for each hierarchy level.</td></tr></table></div>	Flattening mode	Source column	Target column	Vertical	POPULATION	C_POPULATION	Horizontal	POPULATION	C_L1_POPULATION C_L2_POPULATION There is one column for each hierarchy level.
Flattening mode	Source column	Target column								
Vertical	POPULATION	C_POPULATION								
Horizontal	POPULATION	C_L1_POPULATION C_L2_POPULATION There is one column for each hierarchy level.								
<i>Child column</i>	<p>Identifies the column in the source data that contains the child identifier in each parent-child relationship. Drag this column from the source schema into the <i>Child column</i> box.</p> <p>You cannot specify columns that include nested schemas as the child.</p>									
<i>Do not abort in case of cycle</i>	<p>Select to specify that a job should not abort if a cycle (circular dependency) is detected. If a cycle is detected, the software writes warnings to the log file.</p> <p>If left unchecked, the software aborts jobs when the transform encounters a cycle.</p>									

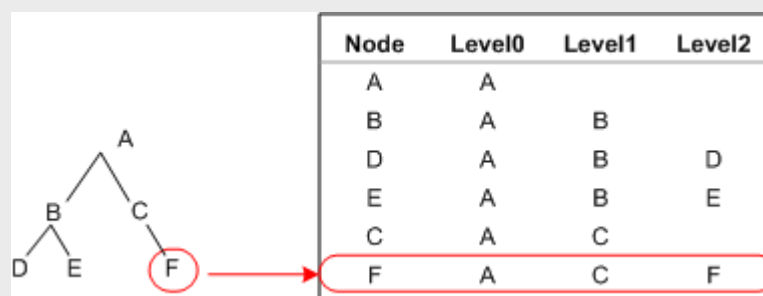
Option	Description
--------	-------------

Flattening type

Indicates how the hierarchical relationships are described in the output. Choose from two options:

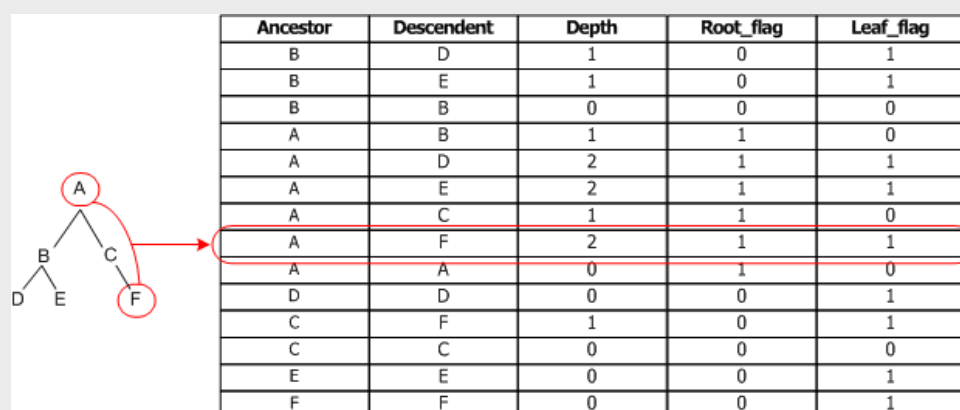
- **Horizontal:** Each row of the output describes a single node in the hierarchy and the path to that node from the root. This mode requires that you specify the maximum path length through the tree as the *Maximum depth*.

❖ Example



- **Vertical:** Each row of the output describes a single relationship between ancestor and descendent and the number of nodes the relationship includes. There is a row in the output for each node and all of the descendants of that node. Each node is considered its own descendent and therefore is listed one time as both ancestor and descendent.

❖ Example



Generate cycle rows

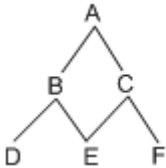

When a cycle is encountered, the circular node is the last in the hierarchy tree, and the tree itself carries a negative value for Leaf Level (Horizontal Flattening) and Depth (Vertical Flattening).

Select to get more information about which nodes are causing the cycles.

❖ Example

insert a Validation transform after the Hierarchy flattening transform to check for negative values. If the software encounters negative values, send the data to another path for further analysis.

Option	Description									
Maximum depth	<p>For horizontal flattening only. Indicates the maximum depth of the hierarchy. The root node (level 0) has a depth of 0; the first level has a depth of 1, and so on.</p> <p>If you do not know the number of levels in your hierarchy, set <i>Maximum depth</i> to 1. When you execute the job, the software issues a warning in the execution log indicating that the <i>Maximum depth</i> is less than the actual depth of the hierarchy. Reset <i>Maximum depth</i> to the actual value reported in the warning message.</p>									
Parent attribute list	<p>Identifies a column or columns that are associated with the parent column. Drag columns from the source schema into the <i>Parent attribute list</i>. The column name appears in the target schema with a prefix that identifies the column as a parent attribute.</p> <div><div><div>❖ Example</div><p>The following table shows the result of adding a column named POPULATION to the parent attribute list.</p><table><tr><th>Flattening mode</th><th>Source column</th><th>Target column</th></tr><tr><td>Vertical</td><td>POPULATION</td><td>P_POPULATION</td></tr><tr><td>Horizontal</td><td>POPULATION</td><td>P_L1_POPULATION P_L2_POPULATION There is one column for each hierarchy level.</td></tr></table></div></div> <p>You can specify columns that include nested schemas as the parent attribute.</p>	Flattening mode	Source column	Target column	Vertical	POPULATION	P_POPULATION	Horizontal	POPULATION	P_L1_POPULATION P_L2_POPULATION There is one column for each hierarchy level.
Flattening mode	Source column	Target column								
Vertical	POPULATION	P_POPULATION								
Horizontal	POPULATION	P_L1_POPULATION P_L2_POPULATION There is one column for each hierarchy level.								
Parent column	<p>Identifies the column in the source data that contains the parent identifier in each parent-child relationship. Drag this column from the source schema into the <i>Parent column</i> box.</p> <p>You cannot specify columns that include nested schemas as the parent attribute.</p>									
Run as separate process	<p>Creates a separate data flow process for the Hierarchy_Flattening transform when Data Services executes the data flow.</p>									

Option	Description
<i>Use maximum length paths</i>	For vertical flattening only. Indicates whether longest or shortest paths are used to describe relationships between descendents and ancestors when the descendent has more than one parent. The option only affects the DEPTH column in the output.
	
The depth of node E is the same for both paths to A	The depth of node M is "1" or "2" depending on the path to J

Parent topic: [Hierarchy_Flattening \[page 380\]](#)

Related Information

[Hierarchy_Flattening transform editor \[page 382\]](#)

[Error conditions \[page 386\]](#)

[Data inputs \[page 387\]](#)

[Data outputs: Horizontal flattening \[page 388\]](#)

[Data outputs: Vertical flattening \[page 390\]](#)

8.4.4.3 Error conditions

If the hierarchy represented by the input data set is cyclic, Data Services produces a run-time error.

A cyclic error occurs when a hierarchy in the input data set has a node that is its own ancestor.

If you select the *Do not abort in case of cycle* option, the software does not produce the runtime error.

If the input data source describes multiple root nodes, the software does not produce errors.

Parent topic: [Hierarchy_Flattening \[page 380\]](#)

Related Information

[Hierarchy_Flattening transform editor \[page 382\]](#)

[Hierarchy_Flattening options \[page 383\]](#)

[Data inputs \[page 387\]](#)

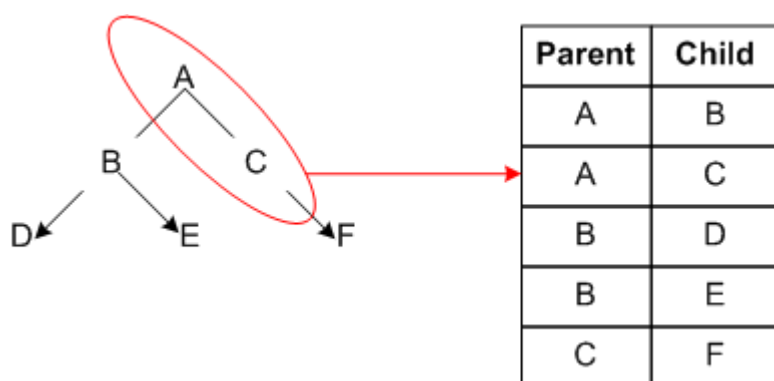
[Data outputs: Horizontal flattening \[page 388\]](#)

[Data outputs: Vertical flattening \[page 390\]](#)

8.4.4.4 Data inputs

Data input for the Hierarchy_Flattening transform includes rows that describe individual parent to child relationships.

The following illustration shows rows in an input table that describe individual parent-child relationships.



Each row must contain two columns that function as the keys of the parent and child in the relationship. The input can also include columns containing attributes that describe the parent and, or child.

❖ Example

Input data set is an Employee Master hierarchy description in which each row represents the relationship between an employee (child node) and the employee manager (parent node). In this example:

- Employ_ID is the child node identifier and the primary key
- Mgr_ID is the parent node identifier

Employee

Employ_ID

Dept

Salary

Mgr_ID

The input data set includes only rows with operation code of NORMAL.

The input data set can contain hierarchical data.

Parent topic: [Hierarchy_Flattening \[page 380\]](#)

Related Information

[Hierarchy_Flattening transform editor \[page 382\]](#)

[Hierarchy_Flattening options \[page 383\]](#)

[Error conditions \[page 386\]](#)

[Data outputs: Horizontal flattening \[page 388\]](#)

[Data outputs: Vertical flattening \[page 390\]](#)

8.4.4.5 Data outputs: Horizontal flattening

Use horizontal flattening so that each row of the output describes a single node in the hierarchy and the path to that node from the root.

Horizontal flattening shows the number of levels in the hierarchy and the distance between a give node and the root node. With horizontal flattening, the root is listed in the first column and the outermost leaf is listed in the last column.

The following table describes the target columns that horizontal flattening produces.

Column name	Description
CURRENT_LEAF	The end node described.
LEAF_LEVEL	The number of levels down from the root node where Current_leaf is found The root node has Leaf_level of 0.
LEVEL0	The descriptor for the top level node.
LEVEL1	The descriptor for the first level node If Leaf_level is 0, this value is NULL.
LEVEL <n>	The descriptor for the <n>th level node where <n> is the number of levels in the hierarchy If Leaf_level is <n> -1 or less, this value is NULL.
P_LO_ <attribute_column>	Parent attribute column associated with the node described in Level0.

Column name	Description
C_L1_<attribute_column>	Child attribute column associated with the node described in Level1 when that node is the child node. If Leaf_level is 0, this value is NULL.
P_L1_<attribute_column>	Parent attribute column associated with the node described in Level1 when that node is the parent node. If Leaf_level is 0, this value is NULL.
C_L<n-1>_<attribute_column>	Child attribute column associated with the child node described in Level <n-1> where <n> is the number of levels in the hierarchy. If Leaf_level is <n> -2, this value is NULL.
P_L<n-1>_<attribute_column>	Parent attribute column associated with the child node described in Level <n-1> where <n> is the number of levels in the hierarchy. If Leaf_level is <n> -2, this value is NULL.
C_L<n>_<attribute_column>	Child attribute column associated with the child node described in Level <n> where <n> is the number of levels in the hierarchy. If Leaf_level is <n> -1, this value is NULL.

Parent topic: [Hierarchy_Flattening \[page 380\]](#)

Related Information

[Hierarchy_Flattening transform editor \[page 382\]](#)

[Hierarchy_Flattening options \[page 383\]](#)

[Error conditions \[page 386\]](#)

[Data inputs \[page 387\]](#)

[Data outputs: Vertical flattening \[page 390\]](#)

8.4.4.5.1 Example of horizontal flattening

Use this example to help you understand horizontal flattening.

❖ Example

The following table shows an example of the target schema and data for horizontal flattening. The target schema contains the following:

- Two level hierarchy
- Country as the root node
- State at level 1 (leaf node)
- City at level 2 (leaf node)

The parent and child attributes are both a population value.

The table shows the three rows in the output. However, the format folds the row data onto two rows. The headings at the top of the table describe the first half of each row of data. The headings at the bottom of the table describe the second half of each row of data.

CUR- RENT_LEAF	LEAF_LEVEL	LEVEL0	LEVEL1	LEVEL2
US	0	US	NULL	NULL
	272,583,805	NULL	NULL	NULL
CA	1	US	CA	NULL
	272,583,805	30,866,851	NULL	NULL
San Fran- cisco	2	US	CA	San Fran- cisco
	272,583,805	30,866,851	30,866,851	723,959
	P_L0_POP	C_L1_POP	P_L1_POP	C_L2_POP

In a typical data flow that includes a Hierarchy_Flattening transform with the same attribute for parent and child nodes, you may follow the transform with a query that filters the duplicated attribute values. The query passes the P_L0_ <attribute_column> and the C_L <n> _ <attribute_column> through as they are. The query keeps either parent or child attribute for the intermediate levels.

8.4.4.6 Data outputs: Vertical flattening

Use vertical flattening so that each row of the output describes a single node in the hierarchy and the path to that node from the root.

The following table shows the target columns that vertical flattening produces.

Column name	Description
ANCESTOR	The node closer to the root node in the relationship described by this row.
DESCENDENT	The node farther from the root node in the relationship described by this row.
DEPTH	Number of levels between the Ancestor and Descendent.
ROOT_FLAG	Identifies the value in the Ancestor column as the top node of the hierarchy. Root_flag is 1 if Ancestor is the root node. Otherwise, Root_flag is 0.

Column name	Description
LEAF_FLAG	Identifies the value in the Descendent column as the bottom node of the hierarchy. Leaf_flag is 1 if Descendent is the leaf node. Otherwise, Leaf_flag is 0.
P_ <attribute_column>	Column from the source that you associate with the parent (can be more than one P_ <attribute_column>). This value is NULL for a row describing the relationship between a leaf node and itself, which means that the Leaf_flag is 1, and the parent and child are the same node.
C_ <attribute_column>	Column from the source that you associate with the child (can be more than one C_ <attribute_column>). This value is NULL for a row describing the relationship between a root node and itself, which means that the Root_flag is 1, and the parent and child are the same node.

Parent topic: [Hierarchy_Flattening \[page 380\]](#)

Related Information

[Hierarchy_Flattening transform editor \[page 382\]](#)

[Hierarchy_Flattening options \[page 383\]](#)

[Error conditions \[page 386\]](#)

[Data inputs \[page 387\]](#)

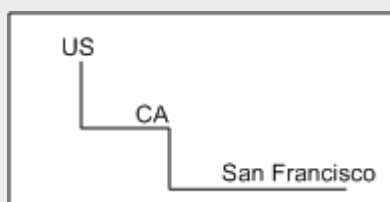
[Data outputs: Horizontal flattening \[page 388\]](#)

8.4.4.6.1 Example of vertical flattening

Use this example to help you understand vertical flattening.

❖ Example

The following graphic and table shows an example of the target schema and data for vertical flattening with a two-level hierarchy. The hierarchy includes a country as the root node, state at level one, and city at level 2 (leaf nodes).



The parent and child attributes are both Sa population value.

ANCESTOR	DESCENDENT	DEPTH	ROOT_FLAG	LEAF_FLAG	P_POP	C_POP
US	CA	1	1	0	272,583,805	30,866,851
US	SanFrancisco	2	1	1	272,583,805	723,959
CA	SanFrancisco	1	0	1	30,866,851	723,959
US	US	0	1	0	272,583,805	NULL
CA	CA	0	0	0	30,866,851	30,866,851
SanFrancisco	SanFrancisco	0	0	1	NULL	723,959


Each node is listed one time as both ancestor and descendent. The Parent attribute is null for a row describing the relationship between a leaf node and itself. Likewise the Child attribute is null for a row describing the relationship between a root node and itself.

The transform ignores any hierarchical data unless a nested schema is identified as a parent or child attribute. An attribute column containing nested data is passed through the transform without change.

8.4.5 History_Preserving

Use the History_Preserving transform to flag rows so that the software preserves the original values in the target.

History_Preserving information

Characteristic	Description
	History_Preserving icon
Use	<p>Produces a new row in your target instead of updating an existing row. You indicate in which columns the transform identifies changes to be preserved.</p> <p>If the value of certain columns change, the transform creates a new row for each row flagged as UPDATE in the input data set.</p>
Data inputs	A data set that is the result of a comparison between two images of the same data in which changed data from the newer image are flagged as UPDATE rows and new data from the newer image are flagged as INSERT rows.
Data outputs	A data set with rows flagged as INSERT or UPDATE.

[History_Preserving transform editor \[page 393\]](#)

When you include the History_Preserving transform in a data flow, complete the applicable options in the editor.

[History_Preserving options \[page 394\]](#)

Specify options in the History_Preserving section to set the software behavior for this transform.

[Example: Data inputs \[page 397\]](#)

Data that is input to the History_Preserving transform contains the result of a comparison of two images of the same data where rows are flagged as UPDATE or INSERT.

[Example: Data outputs \[page 398\]](#)

Output from the History_Preserving transform contains a data set with rows flagged as INSERT or UPDATE.

Parent topic: [Data Integrator transforms \[page 357\]](#)

Related Information

[Data_Transfer \[page 358\]](#)

[Date_Generation \[page 372\]](#)

[Effective_Date \[page 375\]](#)

[Hierarchy_Flattening \[page 380\]](#)

[Key_Generation \[page 401\]](#)

[Map_CDC_Operation \[page 405\]](#)

[Pivot \(Columns to Rows\) \[page 413\]](#)

[Reverse Pivot \(Rows to Columns\) \[page 418\]](#)

[Table_Comparison \[page 422\]](#)

[XML_Pipeline \[page 433\]](#)



8.4.5.1 History_Preserving transform editor

When you include the History_Preserving transform in a data flow, complete the applicable options in the editor.

The History_Preserving transform editor includes the *Schema In* pane, the *Schema Out* pane, and transform options section.

Drag column names from the *Schema In* pane to fill values for the following columns:

- Valid from
- Valid to
- Compare columns
- Current flag Column options

Data Services generates the target schema in response to the values you choose in the transform options. To refresh the target schema after you change the options, choose  [View](#)  [Refresh](#) or press **F5**.

Parent topic: [History_Preserving \[page 392\]](#)

Related Information

[History_Preserving options \[page 394\]](#)

[Example: Data inputs \[page 397\]](#)

[Example: Data outputs \[page 398\]](#)

8.4.5.2 History_Preserving options

Specify options in the History_Preserving section to set the software behavior for this transform.

History_Preserving option descriptions

Option	Description
<i>Compare columns</i>	<p>Specifies the column or columns in the input data set for which this transform compares the before and after images to determine if there are changes.</p> <ul style="list-style-type: none">• If the values in the specified columns in each image of the data match, the transform flags the row as UPDATE. The result updates the warehouse row with values from the new row. The row from the before image is included in the output as UPDATE to effectively update the date and flag information.• If the values in the specified columns in each image do not match, the row from the after image is included in the output of the transform flagged as INSERT. The result adds a new row to the warehouse with the values from the new row. <p>The <i>Compare columns</i> cannot contain nested schemas.</p>
<i>Current flag—Column</i>	<p>Specifies a column from the source schema that identifies the current valid row from a set of rows with the same primary key. Use this flag to indicate whether a row is the most current data in the warehouse for a given primary key.</p> <p>The <i>Current flag—Column</i> value cannot be the same as the <i>Date columns—Valid from</i> option or the <i>Date columns—Valid to</i> option. The <i>Current flag—Column</i> option cannot contain a nested schema.</p> <p>If you specify any one of the above three options, Data Services validates that you specify a <i>Current flag—Column</i> value, <i>Current flag—Set value</i> value, and <i>Current flag—Reset value</i> value.</p>

Option	Description
<i>Current flag—Reset value</i>	<p>Specifies an expression that evaluates to a value with the same data type as the <i>Current flag—Column</i> column.</p> <p>Data Services uses this value to update the <i>Current flag—Column</i> in an existing row in the warehouse that includes changes in one or more of the compare columns.</p> <p>Enter a value for this option to enable the <i>Current flag—reset value</i> option.</p> <p>For added flexibility, you can enter a variable for this option.</p>
<i>Current flag—Set value</i>	<p>Specifies an expression that evaluates to a value with the same data type as the <i>Current flag—Column</i> column.</p> <p>Data Services uses this value to update the <i>Current flag—Column</i> in the new row of the warehouse added to preserve history of an existing row.</p> <p>Enter a value in the <i>Current flag—Column</i> box to enable this option.</p> <p>For added flexibility, you can enter a variable for this option.</p>
<i>Date columns—Valid from</i>	<p>Specifies a date or datetime column from the source schema.</p> <p>If the warehouse uses an effective date to track changes in data, specify a <i>Date columns—Valid from</i> column.</p> <p>Data Services uses this value in the new row in the warehouse added to preserve the history of an existing row. Data Services also uses this value to update the <i>Date columns—Valid to</i> date column in the previously current row in the warehouse.</p> <p>Data Services validates that you have specified a value for both the <i>Date columns—Valid from</i> and the <i>Date columns—Valid to</i> options.</p>
<i>Date columns—Valid to</i>	<p>Specifies a date or datetime column from the source schema.</p> <p>Specify if the warehouse uses an effective date to track changes in data and if you specified a <i>Date columns—Valid from</i> value.</p> <p>This value is used as the new value in the <i>Date columns—Valid to</i> column in the new row added to the warehouse to preserve history of an existing row.</p> <p>The <i>Date columns—Valid to</i> value cannot be the same as the <i>Date columns—Valid from</i> value.</p>

Option	Description
<i>Date columns—Valid to date value</i>	<p>Specify the values to use in the <i>Date columns—Valid to</i> option in the old record and the new record added to the warehouse to preserve history of an existing row.</p> <ul style="list-style-type: none"> • <i>New record</i>: Specify one of the following values: <ul style="list-style-type: none"> ◦ A date value specified as yyyy.mm.dd. The default value is 9000.12.31. ◦ A variable that contains a date value. • <i>Old record</i>: Specify one of the following values: <ul style="list-style-type: none"> ◦ Use the "valid from" date of new record.

❖ Example

The following example shows that the new record, Key 2 *From_Date* contains 2006.01.31 and the old record, Key 1 *To_Date* contains the same value.

Key	Empno	Name	Salary	From_Date	To_Date
1	100	Chang	10000.00	2006.01.01	2006.01.31
2	100	Chang	20000.00	2006.01.31	9000.12.31

- Use a date that is one day before "valid from" date of new record.

❖ Example

The following example shows that the new record, Key 2 *From_Date* contains 2006.01.31 and the old record, Key 1 *To_Date* contains a date that is one day before that value, 2006.01.30.

Key	Empno	Name	Salary	From_Date	To_Date
1	100	Chang	10000.00	2006.01.01	2006.01.30
2	100	Chang	20000.00	2006.01.31	9000.12.31

Option	Description
Preserve delete row(s) as update row(s)	<p>Converts DELETE rows to UPDATE rows in the target warehouse. Also, if you set effective date values in Date columns—Valid from and Date columns—Valid to, sets the Date columns—Valid to value to the jog execution date.</p> <ul style="list-style-type: none"> • Yes: Enables this behavior. • No: Does not enable this behavior. <p>Use this option to maintain slowly changing dimensions by feeding a complete data set first through the Table Comparison transform with its Detect deleted row(s) from comparison table option selected.</p>

Parent topic: [History_Preserving \[page 392\]](#)

Related Information

[History_Preserving transform editor \[page 393\]](#)

[Example: Data inputs \[page 397\]](#)

[Example: Data outputs \[page 398\]](#)

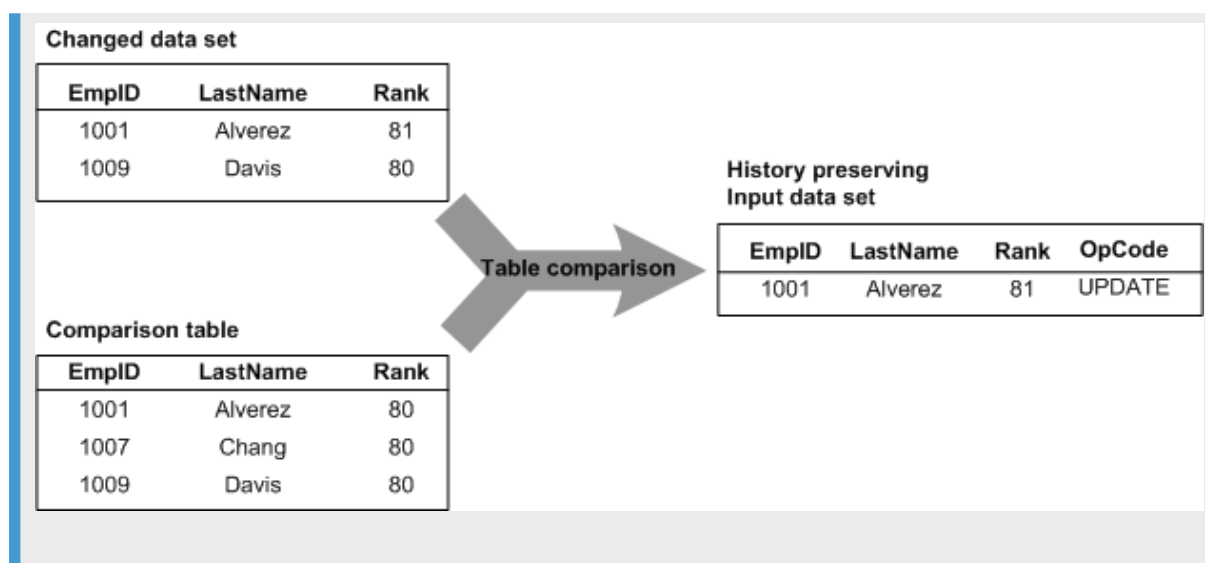
8.4.5.3 Example: Data inputs

Data that is input to the History_Preserving transform contains the result of a comparison of two images of the same data where rows are flagged as UPDATE or INSERT.

Rows that contain changed data from the newer image of the data are flagged as UPDATE. Rows that contain new data from the newer image of the data are flagged as INSERT.

❖ Example

You periodically update A target table that contains employee job rankings information from a source table. The table comparison flags changed data for employee Alvarez and discards unchanged data for employee Davis. The result is a single row flagged with the UPDATE operation code.



The input data set can contain hierarchical data. The transform operates only on the rows at the top-level of the input data set, and passes nested data through to the output without change. Columns containing nested schemas cannot be used as transform parameters.

Use caution when using columns of data type `real` in this transform. Comparison results are unpredictable for this data type.

Parent topic: [History_Preserving \[page 392\]](#)

Related Information

[History_Preserving transform editor \[page 393\]](#)

[History_Preserving options \[page 394\]](#)

[Example: Data outputs \[page 398\]](#)

8.4.5.4 Example: Data outputs

Output from the History_Preserving transform contains a data set with rows flagged as INSERT or UPDATE.

For each row in the input data set, there are two possible outcomes from the transform:

- The software adds an INSERT row.
- The software adds an UPDATE row.

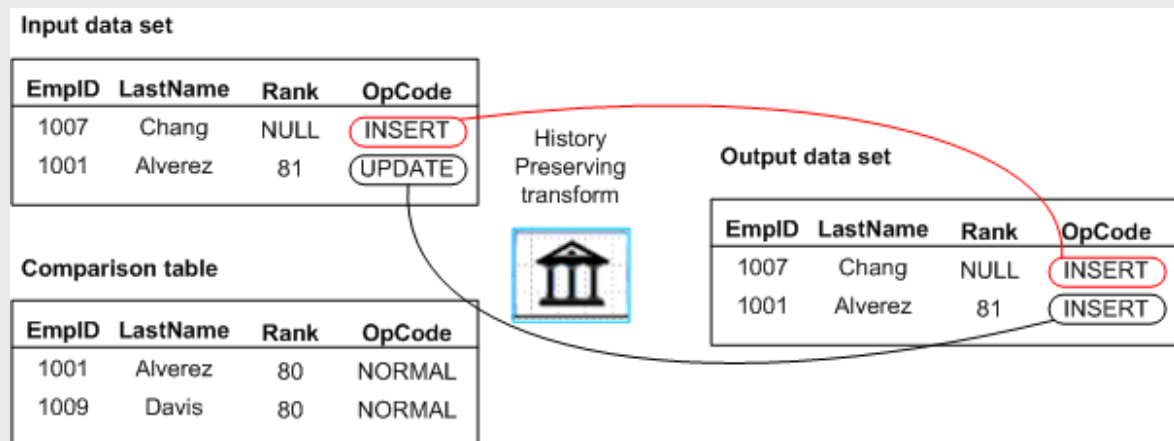
The software adds an INSERT row

An INSERT row happens when the software adds A new row to the comparison table. A new row is required under the following circumstances:

- An input row has an UPDATE flag because a value in a compare column from the input does not match a corresponding value in the comparison table.
- An input row has an INSERT flag because the primary key from the input does not appear in the comparison table.

❖ Example

The transform produces an INSERT row with the values from the input data set row.

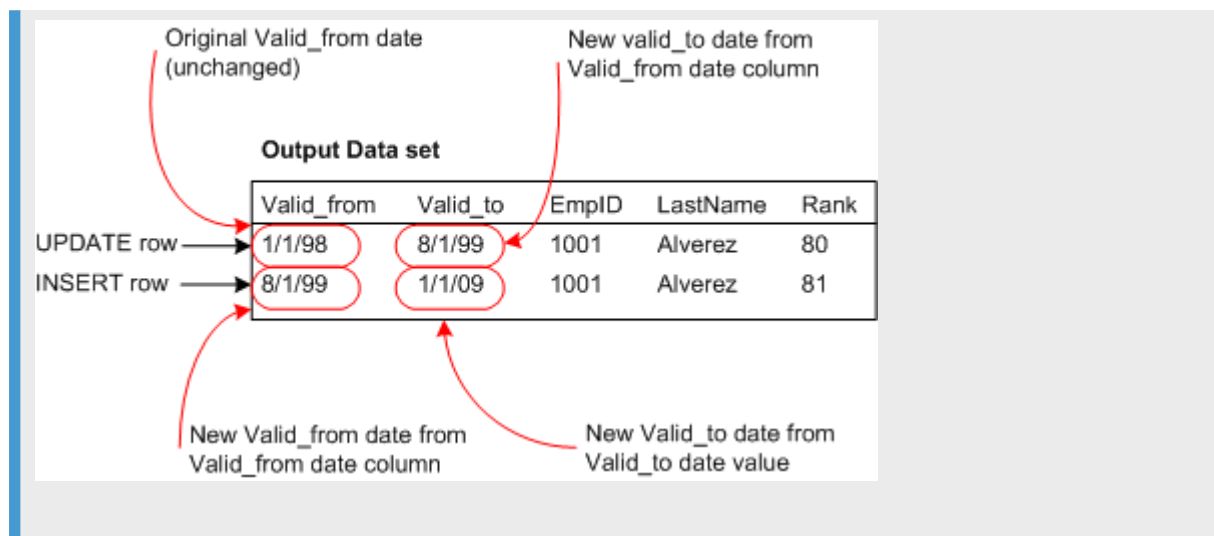


EmpID 1007, Chang is an INSERT row because it is not present in the comparison table.

EmpID 1001, Alvarez is an INSERT row because the row in the Input data set has an UPDATE flag because the Rank changed from 80 to 81.

If you specified flag values for the History_Preserving transform, Data Services includes the Flag set value in the INSERT row. In addition Data Services includes an UPDATE row to update the previously current row in the warehouse with the Flag reset value.

If you specified effective date columns (Valid to date column and Valid from date column), Data Services includes this data as well.

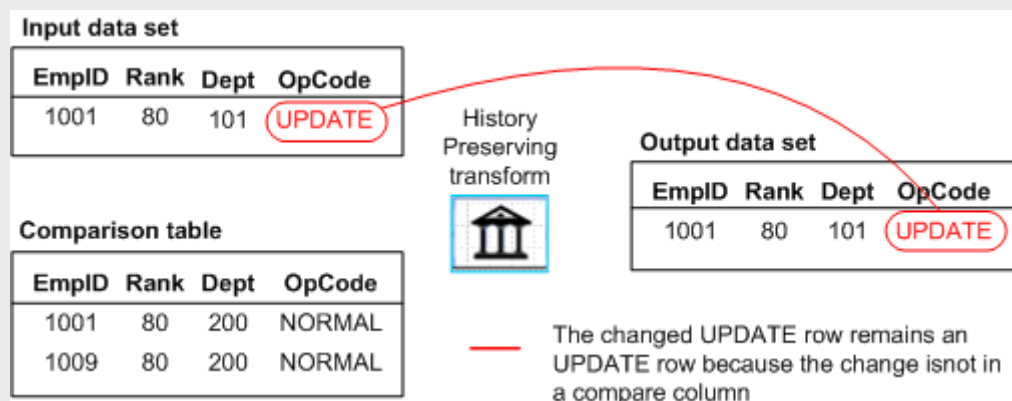


The software adds an UPDATE row

The software adds an UPDATE row under the following circumstances:

- An input row has an UPDATE flag because it contains changes, but not in the compare columns. The transform produces an UPDATE row with the values from the input data set row.

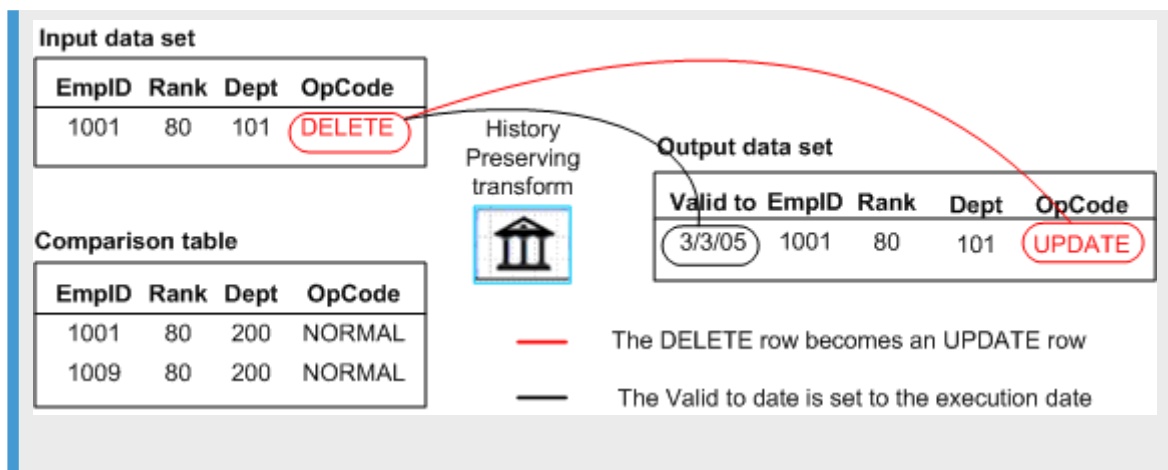
❖ Example



- The input row has a DELETE flag from the input data set row. If you specified the effective date column *Valid to* in the History Preserving transform, Data Services sets its value to the job execution date, and includes the row in the output as an UPDATE row.

❖ Example

Input rows flagged as DELETE contain changes, but not in the compare columns. The transform produces an UPDATE row. The update is that the row is only valid until the Valid to date.



The software passes through nested schemas in the input without change.

Parent topic: [History_Preserving \[page 392\]](#)

Related Information

[History_Preserving transform editor \[page 393\]](#)


[History_Preserving options \[page 394\]](#)

[Example: Data inputs \[page 397\]](#)

8.4.6 Key_Generation

Use the Key_Generation transform to generate new keys for source data that starts from a value based on an existing key in the table.

Key_Generation information

Characteristic	Description
	Key_Generation icon
Use	Generates new keys for new rows in a data set. Generates artificial keys in a table by using the maximum existing key value in the input to use as a starting value for new keys in the output. The transform expects the generated key column to be part of the input schema.

Characteristic	Description
Data inputs	<p>A data set that is the result of a comparison between two images of the same data in which changed data from the newer image are flagged as UPDATE rows and new data from the newer image are flagged as INSERT rows.</p> <p>The data set includes a column into which generated keys are added.</p> <p>The input data set can contain hierarchical data. The transform operates only on the rows at the top-level of the input data set, and passes nested data through to the output without change.</p> <p>Columns containing nested schemas cannot be used as transform parameters.</p>
Data outputs	<p>The input data set with the addition of key values in the generated key column, for input rows flagged as INSERT.</p> <p>Use the Key_Generation transform to produce keys that distinguish rows that would otherwise have the same primary key.</p>

❖ Example

The History_Preserving transform produces rows to add to a warehouse and these rows have the same primary key as rows that already exist in the warehouse. In this case, add a generated key to the warehouse table and fill new key values using the Key_Generation transform.

[Key_Generation transform editor \[page 403\]](#)

When you include the Key_Generation transform in a data flow, you configure the options in the editor.

[Key_Generation options \[page 403\]](#)

Specify options in the Key_Generation section to set the software behavior for this transform.

Parent topic: [Data Integrator transforms \[page 357\]](#)

Related Information

[Data_Transfer \[page 358\]](#)

[Date_Generation \[page 372\]](#)

[Effective_Date \[page 375\]](#)

[Hierarchy_Flattening \[page 380\]](#)

[History_Preserving \[page 392\]](#)

[Map_CDC_Operation \[page 405\]](#)

[Pivot \(Columns to Rows\) \[page 413\]](#)

[Reverse Pivot \(Rows to Columns\) \[page 418\]](#)

[Table_Comparison \[page 422\]](#)

[XML_Pipeline \[page 433\]](#)

8.4.6.1 Key_Generation transform editor

When you include the Key_Generation transform in a data flow, you configure the options in the editor.

The Key_Generation transform editor includes the following sections:

- [Schema In](#) pane on the upper left
- [Schema Out](#) pane on the upper right
- Key_Generation transform options in the lower portion

To refresh the target schema after you change options, choose  [View](#)  [Refresh](#) or press *F5*.

Parent topic: [Key_Generation \[page 401\]](#)

Related Information

[Key_Generation options \[page 403\]](#)

8.4.6.2 Key_Generation options

Specify options in the Key_Generation section to set the software behavior for this transform.

Key_Generation option descriptions

Option	Description
Generated key column	Key column in the source table that contains the existing key value. A column with the same name must exist in the input data set; the software inserts the new keys in this column on output.
Increment value	Interval between generated key values. For added flexibility, enter a variable for this option.

Option	Description																
<i>Table name</i>	<p>Fully qualified name of the source table from which the maximum existing key is determined (key source table). You import the table into the repository before you can use it here. <i>Table name</i> is represented as <code><DATASTORE> . <OWNER> . <TABLE></code> or, if you use Netezza 7.x, it is represented as <code><DATASTORE> . <OWNER> . <SCHEMA> . <TABLE></code>. Where:</p> <ul style="list-style-type: none"> • <code><DATASTORE></code> is the name of the datastore Data Services uses to access the key source table. • <code><OWNER></code>, if required, depends on the associated database type. <p>The following table lists database type and owner values:</p> <table> <tr> <th>Database type</th><th>Owner value</th></tr> <tr> <td>DB2</td><td>Data source dependent</td></tr> <tr> <td>Informix</td><td>Informix-defined user name</td></tr> <tr> <td>Microsoft SQL Server</td><td>User name</td></tr> <tr> <td>ODBC</td><td>Data source dependent</td></tr> <tr> <td>Oracle</td><td>User name</td></tr> <tr> <td>SAP ASE</td><td>User name</td></tr> <tr> <td>SAP Sybase IQ</td><td>User name</td></tr> </table>	Database type	Owner value	DB2	Data source dependent	Informix	Informix-defined user name	Microsoft SQL Server	User name	ODBC	Data source dependent	Oracle	User name	SAP ASE	User name	SAP Sybase IQ	User name
Database type	Owner value																
DB2	Data source dependent																
Informix	Informix-defined user name																
Microsoft SQL Server	User name																
ODBC	Data source dependent																
Oracle	User name																
SAP ASE	User name																
SAP Sybase IQ	User name																

Parent topic: [Key_Generation \[page 401\]](#)


Related Information

[Key_Generation transform editor \[page 403\]](#)

8.4.7 Map_CDC_Operation

Use the Map_CDC_Operation transform to sort input data, map output data, and resolve before- and after-images for UPDATE rows.

Map_CDC_Operation information

Characteristic	Description
	Map_CDC_Operation icon
Use	<p>Performs the following functions when you use the input requirements for the <i>Sequencing column</i> and <i>Row operation column</i>:</p> <ul style="list-style-type: none">• Sorts input data based on values in the <i>Sequencing column</i> dropdown list and optionally the <i>Additional grouping columns</i> box.• Maps output data based on values in the <i>Row operation column</i> dropdown list. The software maps source table rows to INSERT, UPDATE, or DELETE operations before passing them on to the target.• Resolves missing, separated, or multiple before and after images for UPDATE rows.• Allows you to filter columns and view UPDATE rows before running the job. <p>Use the Map_CDC_Operation to support relational or main-frame changed-data capture (CDC). This transform also supports any data stream as long as you meet the input requirements. Relational CDC sources include Oracle and SQL Server.</p> <p>Place this transform at the end of the data flow but before the target because it produces INPUT, UPDATE, and DELETE operation codes. Data Services issues a warning if you use other objects.</p>
Data inputs	<p>All rows in the input data are set to NORMAL. NORMAL is an internal Data Services operation code.</p> <p>The input data set can contain hierarchical data. Nested schemas in the input are passed through without change.</p>

Characteristic	Description
Data outputs	<p>A data set with rows flagged as specified by the values in the column that you selected as the Row operation column.</p> <p>Rows in the input data set all use NORMAL as their internal Data Services operation code.</p> <p>Rows in the output data set can have any of the following operation codes:</p> <ul style="list-style-type: none"> • INSERT • DELETE • UPDATE <p>In addition, the software assigns the DISCARD option under some conditions. The software does not pass through discarded rows to the output of the transform.</p>

[Map_CDC_Operation transform editor \[page 407\]](#)

When you include the Map_CDC_Operation transform in a data flow, you configure the options in the editor.

[Map_CDC_Operation options \[page 407\]](#)

Specify values for the Map_CDC_Operation transform options to set the software behavior for this transform.

[Sorting CDC data \[page 408\]](#)

When you use change data capture (CDC) on target data, ensure that you preserve row order.

[Rules for resolving before and after image pairs \[page 409\]](#)

The Map_CDC_Operation transform has specific rules to resolve before and after image pairs.

[Filtering updated rows \[page 412\]](#)

Apply column filtering so that the transform filters out UPDATE rows on output.

Parent topic: [Data Integrator transforms \[page 357\]](#)

Related Information

[Data_Transfer \[page 358\]](#)

[Date_Generation \[page 372\]](#)

[Effective_Date \[page 375\]](#)

[Hierarchy_Flattening \[page 380\]](#)

[History_Preserving \[page 392\]](#)

[Key_Generation \[page 401\]](#)

[Pivot \(Columns to Rows\) \[page 413\]](#)

[Reverse Pivot \(Rows to Columns\) \[page 418\]](#)

[Table_Comparison \[page 422\]](#)

[XML_Pipeline \[page 433\]](#)

8.4.7.1 Map_CDC_Operation transform editor

When you include the Map_CDC_Operation transform in a data flow, you configure the options in the editor.

The Map_CDC_Operation editor contains the following sections:

- [Schema In](#) in the upper left
- [Schema Out](#) in the upper right
- Transform options in the lower pane

Select the required columns from your input schema to sequence the input rows and map operations to the output. Select the columns from the [CDC Columns](#) dropdown lists.

To sort the input using additional columns, click a column in the [Schema In](#) pane and drag it into the [Additional grouping columns](#) box.

If you are using a relational or mainframe CDC source table, the DI_SEQUENCE_NUMBER and DI_OPERATION_TYPE columns appear in the input schema. The software automatically removes them in the output schema. To propagate the DI_SEQUENCE_NUMBER and the DI_OPERATION_TKYPE columns to the output, create additional columns that map to them in the input data to the Map_CDC_Operation transform.

Parent topic: [Map_CDC_Operation](#) [page 405]

Related Information

[Map_CDC_Operation options](#) [page 407]

[Sorting CDC data](#) [page 408]

[Rules for resolving before and after image pairs](#) [page 409]

[Filtering updated rows](#) [page 412]

8.4.7.2 Map_CDC_Operation options

Specify values for the Map_CDC_Operation transform options to set the software behavior for this transform.

Map_CDC_Operation option descriptions

Option	Description
Additional grouping columns	In addition to the Sequencing column , you can sort input on additional columns by dragging them into this box from the input schema. Sorts are prioritized based first on the sequencing column and then on the order of the columns added to this box.

Option	Description
Define columns to filter updated rows	Drag and drop columns from the input schema to this box to compare CDC updated rows in the before and after images. If the values in the columns differ between the before and after images, Data Services generates an UPDATE row, otherwise the rows are filtered out.
Input already sorted by sequencing column	<p>By default, the transform assumes that you have already sorted the input data based on the value selected in the Sequencing column box. If you deselect this checkbox, Data Services re sorts the input data using the value in the Sequencing column box.</p> <p>Use the re sort capability of this transform only when necessary as it impacts job performance.</p>
Row operation column	<p>Required. Specifies a column with one of the following output operation codes for each row:</p> <ul style="list-style-type: none"> • I for INSERT • B for before-image of an UPDATE • U for after-image of an UPDATE • D for DELETE <p>If you are using a relational or mainframe CDC source table, the <code>DI_OPERATION_TYPE</code> column is automatically selected as the Row operation column.</p>
Sequencing column	<p>(Required) Specifies an integer used to order table rows.</p> <p>If you are using a relational or mainframe CDC source table, the <code>DI_SEQUENCE_NUMBER</code> column is automatically selected as the Sequencing column.</p>

Parent topic: [Map_CDC_Operation \[page 405\]](#)

Related Information

[Map_CDC_Operation transform editor \[page 407\]](#)

[Sorting CDC data \[page 408\]](#)

[Rules for resolving before and after image pairs \[page 409\]](#)

[Filtering updated rows \[page 412\]](#)

8.4.7.3 Sorting CDC data

When you use change data capture (CDC) on target data, ensure that you preserve row order.

The following example illustrates why preserving row order is important.

❖ Example

The following operations are applied to an empty target:

- `INSERT into TAB1 values ('Bob', 'Boat', 3500)`
- `INSERT into TAB1 values ('Jane', 'BMW Roadster', 24000)`
- `UPDATE TAB1 set toy = 'Motorcycle', price = 12000 where name = 'Bob'`
- `DELETE from TAB1 where name = 'Bob'`

The table TAB1 is left with one row:

```
'Jane', 'BMW Roadster', 24000
```

If these operations are applied out of order, for example, if the DELETE occurs before the UPDATE operation, then database consistency is no longer preserved. In this example:

- The table has two rows (Bob and Jane)
- The last UPDATE statement fails because there is no row on which to perform an UPDATE

By ordering the input rows using the sequencing column, you preserve the order of the original set of operations.

The sequencing column values are also useful. If you use before and after images for update rows, it is possible that the pairs may be separated, multiplied, or lost depending on the design of your data flow. You can re sort input columns as needed by using the sequencing column and any number of additional columns.

The before and after images of an UPDATE row have the same sequence value. Thus correctly sorted before and after image rows are listed in pairs.

Parent topic: [Map_CDC_Operation \[page 405\]](#)

Related Information

[Map_CDC_Operation transform editor \[page 407\]](#)

[Map_CDC_Operation options \[page 407\]](#)

[Rules for resolving before and after image pairs \[page 409\]](#)

[Filtering updated rows \[page 412\]](#)

8.4.7.4 Rules for resolving before and after image pairs

The Map_CDC_Operation transform has specific rules to resolve before and after image pairs.

The Map_CDC_Operation transform uses the following rules to process and resolve before and after images:

- When constructing UPDATE rows, the transform uses the value in *Row Operation Column*. If there are before images in the input stream, the transform combines the before (B) and after image (U) row pairs into one UPDATE row.

❖ Example

Given the following sample input of six rows:

Sequencing Column	Operation Column	Internal Operation Code
1	I	NORMAL
2	B	NORMAL
2	U	NORMAL
3	D	NORMAL
4	B	NORMAL
4	U	NORMAL

The transform outputs the following four rows:

Sequencing Column	Operation Column	Internal Operation Code
1	I	INSERT
2	U	UPDATE (before and after images)
3	D	DELETE
4	U	UPDATE (before and after images)

- If there are no before images (B) in the input stream, the after images (U) alone produce UPDATE rows.

❖ Example

Given the following sample input rows:

Sequencing Column	Operation Column	Internal Operation Code
1	I	NORMAL
2	U	NORMAL
3	D	NORMAL
4	U	NORMAL

The transform outputs the following four rows:

Sequencing Column	Operation Column	Internal Operation Code
1	I	INSERT
2	U (no before)	UPDATE
3	D	DELETE
4	U (no before)	UPDATE

- If a before image (B) row is followed by additional B rows, the subsequent B rows are ignored until an after image (U) row is encountered.

❖ Example

Given the following six input rows:

Sequencing Column	Operation Column	Internal Operation Code
1	U	NORMAL
1	B	NORMAL
2	B	NORMAL
3	B	NORMAL
4	B	NORMAL
2	U	NORMAL

The transform outputs the following two UPDATE rows:

Sequencing Column	Operation Column	Internal Operation Code
1	U	UPDATE (before and after images)
2	U	UPDATE (before and after images)

- The first two rows are processed as one UPDATE row.
- The third and sixth rows are processed as a pair. One UPDATE row is output.
- The remaining rows are DISCARDED. For more information, see the following rule.

- If after a series of B rows, either no U rows remain or another row type is encountered, B rows are discarded.

❖ Example

Given the following sample input of five rows:

Sequencing Column	Operation Column	Internal Operation Code
1	U	NORMAL
1	B	NORMAL
2	B	NORMAL
3	B	NORMAL
4	I	NORMAL

The transform outputs the following two rows

Sequencing Column	Operation Column	Internal Operation Code
1	U	UPDATE (before and after images)
4	I	INSERT

Parent topic: [Map_CDC_Operation \[page 405\]](#)

Related Information

[Map_CDC_Operation transform editor \[page 407\]](#)

[Map_CDC_Operation options \[page 407\]](#)

[Sorting CDC data \[page 408\]](#)

[Filtering updated rows \[page 412\]](#)

8.4.7.5 Filtering updated rows

Apply column filtering so that the transform filters out UPDATE rows on output.

To filter change data capture (CDC) updated rows, drag and drop columns from the input schema to the [Define columns to filter updated rows](#) box.

Filter criteria: If two input rows have the same sequencing column value and operation column value (B and U), the Map_CDC_Operation transform compares the before image and after image of the selected columns in the column filter.

If the values in the columns in the filter differ between the before image and after image, Data Services generates an UPDATE row; otherwise, the row is filtered out. If there is no column filter, all updated rows coming from the input stream are passed through.

LONG, BLOB, and columns selected in [Sequencing column](#) and [Row operation column](#) options are not allowed in the [Define columns to filter updated rows](#) list.

i Note

Because the [Define columns to filter updated rows](#) option filters out CDC updated rows, a target table should have a primary key that identifies the row to update the change record.

Parent topic: [Map_CDC_Operation \[page 405\]](#)

Related Information

[Map_CDC_Operation transform editor \[page 407\]](#)

[Map_CDC_Operation options \[page 407\]](#)


[Sorting CDC data \[page 408\]](#)

[Rules for resolving before and after image pairs \[page 409\]](#)

8.4.8 Pivot (Columns to Rows)

Use the Pivot transform to rotate the values in specified columns to rows.

Pivot information

Characteristic	Description
	Pivot transform icon
Use	<p>Creates a new row for each value in a column that you identify as a pivot column.</p> <p>The Pivot transform allows you to change how the relationship between rows is displayed. For each value in each pivot column, Data Services produces a row in the output data set. You can create pivot sets to specify more than one pivot column.</p>
Data inputs	A data set with rows flagged as NORMAL.
Data outputs	A data set with rows flagged as NORMAL. This target includes the nonpivoted columns, a column for the sequence number, the data field column, and the pivot header column.

[Pivot transform options \[page 414\]](#)

Specify options in the Pivot transform options section to set the software behavior for this transform.

[Pivot transform examples \[page 415\]](#)

Examples that illustrate how to use the Pivot transform.

Parent topic: [Data Integrator transforms \[page 357\]](#)

Related Information

[Data_Transfer \[page 358\]](#)

[Date_Generation \[page 372\]](#)

[Effective_Date \[page 375\]](#)

[Hierarchy_Flattening \[page 380\]](#)

[History_Preserving \[page 392\]](#)

[Key_Generation \[page 401\]](#)

[Map_CDC_Operation \[page 405\]](#)

[Reverse Pivot \(Rows to Columns\) \[page 418\]](#)

[Table_Comparison \[page 422\]](#)

8.4.8.1 Pivot transform options

Specify options in the Pivot transform options section to set the software behavior for this transform.

Pivot transform option descriptions

Option	Description
<i>Data field column</i>	<p>Name of the column that contains the pivoted data. This column contains all of the values in <i>Pivot columns</i>.</p> <p>The data type of this column is determined by the data type set in <i>Pivot columns</i>. If two or more <i>Pivot columns</i> contain different data types, Data Services converts the columns to a single data type; the data type of the first column you add to the pivot set.</p>
<i>Header column</i>	Name of the column that contains the pivoted column names. This column lists the names of the columns where the corresponding data originated.
<i>Non-pivot columns</i>	Columns in the source that are to appear in the target without modification.
<i>Pivot columns</i>	Set of columns to be rotated into rows. Describe these columns in the <i>Header column</i> . Describe the data in these columns in the <i>Data field column</i> .
<i>Pivot sequence column</i>	<p>Name you assign to the sequence number column. For each row created from a pivot column, Data Services increments and stores a sequence number. Data Services resets the sequence to 1 when creating a row from an original row.</p> <div> <p>❖ Example</p> <p>For example, if the row corresponds to the first column pivoted, the sequence number for the row is 1.</p> </div>
<i>Pivot set</i>	Number that identifies a pivot set. For each pivot set, you define a group of pivot columns, a pivot data field, and a pivot header name. Each pivot set has a unique <i>Data field column</i> and <i>Header column</i> . Data Services automatically saves this information.

Parent topic: [Pivot \(Columns to Rows\) \[page 413\]](#)

Related Information

[Pivot transform examples \[page 415\]](#)

8.4.8.2 Pivot transform examples

Examples that illustrate how to use the Pivot transform.

In the first example, your source data is a table that contains rows for expenses, broken down by expense type.

Emp_name	Mgr_ID	Internal_Expense	Travel_Expense	Misc_Expense
AAA	1234	2000.00	5000.00	100.00
BBB	9876	3000.00	0.00	1000.00
CCC	5555	4800.00	800.00	0.00

This source table has expense numbers in several columns, which makes it difficult for you to calculate expense summaries. The Pivot transform rearranges the data into a more manageable form, with all expenses in a single column, without losing category information.

Set the Pivot transform options to pivot the data so that all of the expenses are in the same column. You only need one pivot set in this case. Set the Pivot transform options as shown in the following table.

Options	Value
Pivot sequence column	Sequence
Nonpivot columns	Emp_name
Pivot set	1
Data field column	Expense
Header column	Expense_Type
Pivot columns	Internal_Expense Travel_Expense Misc_Expense

Select [ViewRefresh](#) or press **F5** to update the output schema for the Pivot transform.

The output data set includes the employee name (not pivoted) and new columns for the pivot sequence, expense type (pivot header), and actual expense data. The manager ID column is not listed in either the pivot or the nonpivot column lists, so it is not included in the output.

The result is a single column of expense values that you can easily summarize.

Emp_name	Sequence	Expense_Type	Expense
AAA	1	Internal_Expense	2000.00
AAA	2	Travel_Expense	5000.00
AAA	3	Misc_Expense	100.00

Emp_name	Sequence	Expense_Type	Expense
BBB	1	Internal_Expense	3000.00
BBB	2	Travel_Expense	0.00
BBB	3	Misc_Expense	1000.00
CCC	1	Internal_Expense	4800.00
CCC	2	Travel_Expense	800.00
CCC	3	Misc_Expense	0.00

For another example, your source table now contains two types of data spread out over two months: Expenses and days traveling both domestically and internationally.

Emp_name	Dom_Exp	Int_Exp	Dom_Day	Int_Day
AAA	2000.00	5000.00	10	5
BBB	3000.00	0.00	0	0
CCC	4800.00	800.00	15	1

To create a target table that has the data in two columns, expenses and days, you create two pivot sets. Create one pivot set to pivot on the expense columns by making the settings shown in the following table.

Options	Value
Pivot sequence column	Seq
Nonpivot columns	Emp_name
Pivot set	1
Data field column	Expense
Header column	Expense_Type
Pivot columns	Dom_Exp Int_Exp

Create a second pivot set to pivot on the day columns using the values in the following table.

Options	Value
Pivot sequence column	Seq

Options	Value
Nonpivot columns	Emp_name
Pivot set	2
Data field column	Num_Days
Header column	Day_Type
Pivot columns	Dom_Day Int_Day

The result is that the output data set includes the employee name (not pivoted), and the following new columns:

- Pivot sequence
- Expense type
- Expense data
- Day type
- Day data

Because you linked the pivot data, domestic and international data is contained in unique rows.

Emp_name	Seq	Expense_Type	Expenses	Day_Type	Num_Days
AAA	1	Dom_Exp	2000.00	Dom_Day	10
AAA	2	Int_Exp	5000.00	Int_Day	5
BBB	1	Dom_Exp	3000.00	Dom_Day	12
BBB	2	Int_Exp	0.00	Int_Day	0
CCC	1	Dom_Exp	4800.00	Dom_Day	15
CCC	2	Int_Exp	800.00	Int_Day	1

When working with multiple pivot sets, pivoted columns cannot contain a different number of rows.

If the example target table contained additional expenses (internal plus miscellaneous expenses), but only had days traveled to match domestic and international travel expenses, the expense data set would be larger than the days traveled data set. In that case, you would have to add a new artificial column containing NULL values to the input data set, and associate the day columns with those additional expenses.

Emp_name	Seq	Expense_Type	Expenses	Day_Type	Num_Days
AAA	1	Dom_Exp	2000.00	Dom_Day	10
AAA	2	Int_Exp	5000.00	Int_Day	5

Emp_name	Seq	Expense_Type	Expenses	Day_Type	Num_Days
AAA	3	Internal_Exp	500.00	NULL	NULL
AAA	4	Misc_Exp	75.00	NULL	NULL
BBB	1	Dom_Exp	3000.00	Dom_Day	12
BBB	2	Int_Exp	0.00	Int_Day	0
BBB	3	Internal_Exp	350.00	NULL	NULL
BBB	4	Misc_Exp	140.00	NULL	NULL
CCC	1	Dom_Exp	4800.00	Dom_Day	15
CCC	2	Int_Exp	800.00	Int_Day	1
CCC	3	Internal_Exp	1000.00	NULL	NULL
CCC	4	Misc_Exp	55.00	NULL	NULL

Parent topic: [Pivot \(Columns to Rows\) \[page 413\]](#)


Related Information

[Pivot transform options \[page 414\]](#)

8.4.9 Reverse Pivot (Rows to Columns)

Use the Reverse Pivot to rotate the values in specified rows to columns.

Reverse Pivot information

Characteristic	Description
	Reverse Pivot icon

Characteristic	Description
Use	<p>Creates one row of data from several existing rows.</p> <p>The Reverse Pivot transform allows you to combine data from several rows into one row by creating new columns. For each unique value in a pivot axis column and each selected pivot column, Data Services produces a column in the output data set.</p>
Data inputs	A data set with rows flagged as NORMAL.
Data outputs	A data set with rows flagged as NORMAL. This target includes the nonpivoted columns and a column for the combination of each pivot column and each pivot axis.

[Reverse Pivot transform options \[page 420\]](#)

Specify options in the Reverse Pivot transform editor to set the software behavior for this transform.

[Reverse Pivot examples \[page 420\]](#)

Examples that illustrate how to use the Reverse Pivot transform.

Parent topic: [Data Integrator transforms \[page 357\]](#)

Related Information

[Data_Transfer \[page 358\]](#)

[Date_Generation \[page 372\]](#)

[Effective_Date \[page 375\]](#)

[Hierarchy_Flattening \[page 380\]](#)

[History_Preserving \[page 392\]](#)

[Key_Generation \[page 401\]](#)

[Map_CDC_Operation \[page 405\]](#)

[Pivot \(Columns to Rows\) \[page 413\]](#)

[Table_Comparison \[page 422\]](#)

[XML_Pipeline \[page 433\]](#)

8.4.9.1 Reverse Pivot transform options

Specify options in the Reverse Pivot transform editor to set the software behavior for this transform.

Reverse Pivot option descriptions

Option	Description
<i>Axis value</i>	The value of the pivot axis column that represents a particular set of output columns. A set of <i>Pivoted columns</i> is generated for each axis value. There should be one <i>Axis value</i> for each unique value in the <i>Pivot axis column</i> .
<i>Column Prefix</i>	Text added to the front of the <i>Pivoted column</i> names when creating new column names for the rotated data. An underscore separates the prefix name from the pivoted column name.
<i>Default value</i>	The value stored when a rotated column has no corresponding data. If you do not enter a value, the default is "null". Do not enter a blank.
<i>Duplicate value</i>	Action taken when a collision occurs. A collision occurs when there is more than one row with the same key and value in the <i>Pivot axis column</i> . In this case, you can store either the first row or the last row, or you can abort the transform process.
<i>Input data is grouped</i>	Select to indicate whether you have already sorted the input rows based on columns specified in the <i>Non pivot columns</i> box. Presorting can improve the performance of the transform.
<i>Non-pivot columns</i>	The columns in the source table that will appear in the target table without modification.
<i>Pivot axis column</i>	The column that determines what new columns are needed in the output table. At runtime, the transform creates a new column for each <i>Pivoted column</i> and each unique value in this column.
<i>Pivoted columns</i>	The columns containing data you want rotated into the same row. A set of columns is created for each unique value in the <i>Pivot axis column</i> .

Parent topic: [Reverse Pivot \(Rows to Columns\) \[page 418\]](#)

Related Information

[Reverse Pivot examples \[page 420\]](#)

8.4.9.2 Reverse Pivot examples

Examples that illustrate how to use the Reverse Pivot transform.

Suppose that you have a table containing contact information for each employee. Each row in the table contains data for a particular employee and contact type.

EmpNo	Type	Name	Address	Phone
100	emergency	Andrew	404 Hallam St	555-4450
100	home	Pat	125 Mercury St	555-6035

EmpNo	Type	Name	Address	Phone
100	work	Sean	8400 Page Mill Rd	555-5000
200	emergency	Linda	126 River Rd	555-1087
200	home	David	479 Mill St	555-6914
300	work	Joanne	9500 Page Mill Rd	555-8500

Because the table can have several rows for each employee, finding information, such as a missing contact, for a particular employee may be difficult. The Reverse_Pivot transform can rearrange the data into a more searchable form without losing the category information.

Set the Reverse_Pivot transform options to pivot the data such that all of the contact information for a particular employee is in the same row.

Option	Value
Non-pivot columns	EmpNo
Pivoted columns	Name Phone
Default value	Null
Pivot axis column	Type
Duplicate value	Abort
Axis Value	emergency home work
Column Prefix	emergency home work

The output data set includes the employee number field (not pivoted) and two fields—name and phone—for each pivot axis. In this case, there are three pivot axes (emergency, home, and work). Therefore, there are six additional fields. In cases where there is no data for a field in the initial source, the Reverse_Pivot transform stores a null value.

The result is a single row for each employee, which you can use to search easily for missing contact information.

EmpNo	Emerg_Name	Emerg_Phone	Home_Name	Home_Phone	Work_Name	Work_Phone
100	Andrew	555-4450	Pat	555-6035	Sean	555-5000
200	Linda	555-1087	David	555-6914	Null	Null
300	Null	Null	Null	Null	Joanne	555-8500

Parent topic: [Reverse Pivot \(Rows to Columns\) \[page 418\]](#)


Related Information

[Reverse Pivot transform options \[page 420\]](#)

8.4.10 Table_Comparison

Use the Table_Comparison transform to compare two data sets and produce the difference between them as a data set with rows flagged as INSERT and UPDATE.

Table_Comparison information

Characteristic	Description
	Table_Comparison icon
Use	<p>Compares two data sets and produces the difference between them as a data set with rows flagged as INSERT, UPDATE, or DELETE.</p> <p>The Table_Comparison transform allows you to detect and forward changes that have occurred since the last time a target was updated.</p> <div><p>i Note</p><p>To use the Table_Comparison transform with Teradata versions 13 and later as the comparison or target table, perform the following tasks:</p><ul style="list-style-type: none">On the Teradata server, set the <i>General</i> parameter <i>DBSControl</i> to TRUE to allow uncommitted data to be read.In the Data Services Teradata datastore, add the following statement in the "Additional session parameters" field:<pre>SET SESSION CHARACTERISTICS AS TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;</pre></div>

Characteristic	Description
Data inputs	<p>There are two types of data inputs:</p> <ul style="list-style-type: none"> Input data set: The data set from a source or the output from another transform. Only rows flagged as NORMAL are considered by the transform. Comparison table: The specification for a database table to compare to the input data set. <p>If the input data set contains hierarchical (nested) data, Data Services includes only the top-level data in the comparison and does not pass nested schemas through to the output.</p> <p>Use caution when using columns of data type <code>real</code> in this transform. Comparison results are unpredictable for this data type.</p>
Data outputs	<p>A data set containing rows flagged as INSERT, UPDATE, or DELETE. This data set contains only the rows that make up the difference between the two input sources: one from the input to the transform (input data set), and one from a database table you specify in the transform (the comparison table). The transform selects rows from the comparison table based on the primary key values from the input data set. The transform compares columns that exist in the schemas for both inputs. For more information about data outputs, see Data outputs [page 424].</p>

[Data outputs \[page 424\]](#)

Data outputs for the Table_Comparison transform include the rows that make up the difference between the two input sources.

[INSERT row data output \[page 425\]](#)

The software may output an INSERT row.

[UPDATE row data output \[page 426\]](#)

The software may output an UPDATE row.

[Table_Comparison options \[page 427\]](#)

The Table_Comparison editor contains options that you complete based on the type of data that might have changed since you last updated the target warehouse.

Parent topic: [Data Integrator transforms \[page 357\]](#)

Related Information

[Data_Transfer \[page 358\]](#)

[Date_Generation \[page 372\]](#)

[Effective_Date \[page 375\]](#)

[Hierarchy_Flattening \[page 380\]](#)
[History_Preserving \[page 392\]](#)
[Key_Generation \[page 401\]](#)
[Map_CDC_Operation \[page 405\]](#)
[Pivot \(Columns to Rows\) \[page 413\]](#)
[Reverse Pivot \(Rows to Columns\) \[page 418\]](#)
[XML_Pipeline \[page 433\]](#)
[Teradata \[page 125\]](#)

8.4.10.1 Data outputs

Data outputs for the Table_Comparison transform include the rows that make up the difference between the two input sources.

A data set containing rows flagged as INSERT, UPDATE, or DELETE. This data set contains only the rows that make up the difference between the two input sources:

- From the input data set to the transform
- From the comparison table to the transform

The transform selects rows from the comparison table based on the primary key values from the input data set. The transform compares columns that exist in the schemas for both inputs.

If a column has a `date` data type in one table and a `datetime` data type in the other table, the transform compares only the date sections of the data. The columns can also be `time` and `datetime` data types, in which case Data Services only compares the time sections of the data.

The transform generates a data set consisting of rows with INSERT and UPDATE operation codes. However, if you use the *Detect Deleted row(s) from comparison table* option, Data Services produces DELETE rows. If a primary key value in the comparison table is not present in the input data set, no corresponding row appears in the output.

For each row in the input data set, there are four possible outputs from the transform:

- INSERT row
- UPDATE row
- DELETE row
- Ignored row

Parent topic: [Table_Comparison \[page 422\]](#)

Related Information

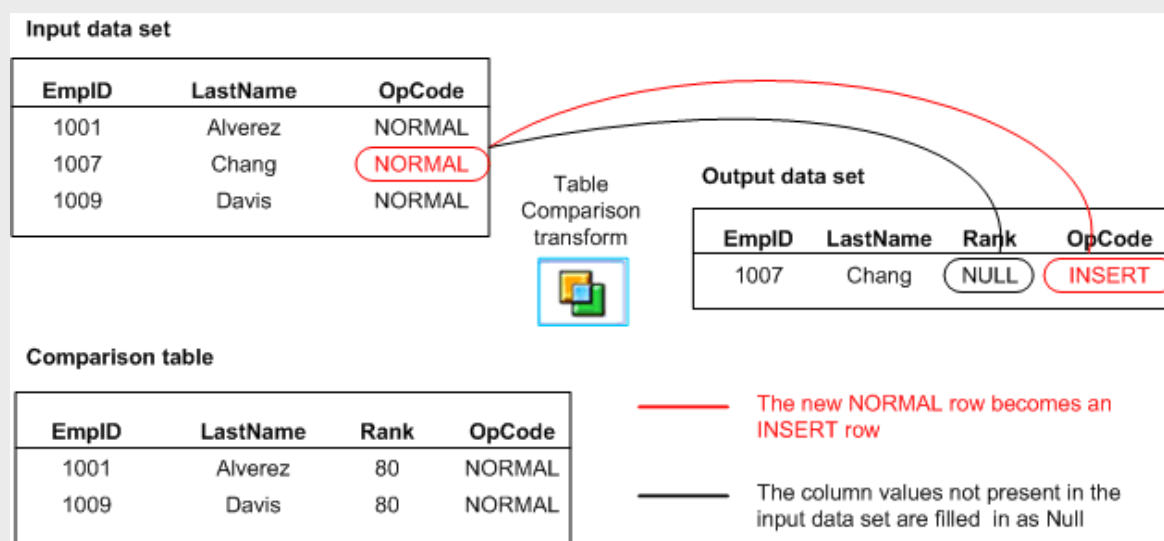
[INSERT row data output \[page 425\]](#)
[UPDATE row data output \[page 426\]](#)
[Table_Comparison options \[page 427\]](#)

8.4.10.2 INSERT row data output

The software may output an INSERT row.

When the primary key value from the input data set does not match a value in the comparison table, the transform produces an INSERT row with the values from the input data set row. If there are columns in the comparison table that are not present in the input data set, the transform adds these columns to the output schema and fills them with NULL values.

❖ Example



i Note

Data Services ignores trailing blanks when it compares values in the comparison table and the input data set. However, an Oracle database server includes trailing blanks in comparisons. Therefore, the Table_Comparison transform produces an INSERT row under the following circumstances:

- The comparison table is an Oracle table with data that had trailing blanks removed.
- The input data contains trailing blanks.
- You use the *Row-by-row select* comparison method. This method pushes down the comparison to the Oracle database server.

To avoid inserting rows when the data differs only by number of trailing blanks, take either of the following actions:

- Use a different comparison method (*Cached comparison table* or *Sorted input*) if possible
- Add the rtrim or rtrim_blank function to remove trailing blanks from the input data.

Parent topic: [Table_Comparison](#) [page 422]

Related Information

[Data outputs \[page 424\]](#)

[UPDATE row data output \[page 426\]](#)

[Table_Comparison options \[page 427\]](#)

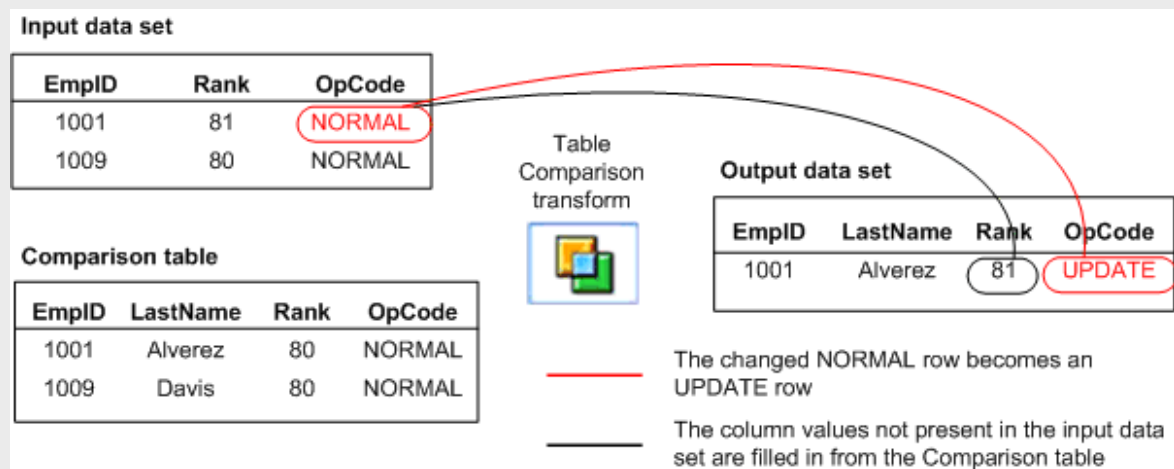
8.4.10.3 UPDATE row data output

The software may output an UPDATE row.

When the primary key value from the input data set matches a value in the comparison table, and values in the non-key compare columns differ in the corresponding rows from the input data set and the comparison table, the transform produces an UPDATE row. The UPDATE row contains the values from the input data set row.

If there are columns in the comparison table that are not present in the input data set, the transform adds these columns to the output schema and fills them with values from the comparison table.

❖ Example



i Note

Data Services ignores trailing blanks when it compares values in the comparison table and the input data set. Therefore, if the input data and the value in the comparison table differ only by trailing blanks, the transform will not produce an UPDATE row.

Parent topic: [Table_Comparison \[page 422\]](#)

Related Information

[Data outputs \[page 424\]](#)

[INSERT row data output \[page 425\]](#)

[Table_Comparison options \[page 427\]](#)

8.4.10.4 Table_Comparison options

The Table_Comparison editor contains options that you complete based on the type of data that might have changed since you last updated the target warehouse.

To help you decide which options to complete, consider whether individual records have changed more than once. For example, do duplicate primary key rows exist or might they be encountered during job processing? If records have changed more than once, decide if you want to record all changes, or just the latest change.

If the columns you specify for the *Input primary keys* option have a unique key per row, then you do not need to use the *Input contains duplicate keys* option.

If a record (row) might change more than once during job execution, consider a plan to handle duplicate keys:

- If you assume that the input data set has duplicate keys, then to avoid data corruption, select the *Input contains duplicate keys* option. This option allows the Table_Comparison transform to output all duplicate key rows in the input data set columns.
- If you assume that the comparison table has duplicate keys, then to avoid data corruption, use the *Generated Key column* option. This option reads only the largest in a set of duplicated keys.
- If you use the *Generated Key column* option with the *Detect Deleted row(s) from comparison table* option, you can specify whether to detect deletes in all duplicate rows or in the row with the largest generated key.

The following table provides more detail about the Table_Comparison options.

Option	Description
Compare columns	<p>Optional. Improves performance. Compares only the subset of columns that you drag into this box from the input schema. If you do not list any columns, the software uses all input columns that are also in the comparison table as compare columns. Not applicable for Long or Blob data types, or the Generated key column.</p> <p>Do not add primary key columns to the compare column list. The transform compares primary key columns before the compare columns. The compare columns apply only if the primary key value from the input data set matches a value in the comparison table.</p> <p>If the primary key value from the input data set does not match a value in the comparison table, Data Services generates an INSERT row and stops comparisons. Data Services generates an UPDATE row with the values from the input data set row when:</p> <ul style="list-style-type: none"> • The primary key value from the input data set matches a value in the comparison table. • Values in the non key compare columns differ in the corresponding rows from the input data set and the comparison table.
Comparison method	<p>Method for accessing the comparison table:</p> <ul style="list-style-type: none"> • Row-by-row select: Looks up the target table using SQL every time it receives an input row. Consider this option when the target table is large compared to the number of rows input to the transform. • Cached comparison table: Loads the comparison table into memory. Queries the comparison table access memory rather than the actual table. Consider this option when you compare the entire target table. Data Services uses pageable cache as the default. If the table fits in the available memory, change the cache type to in-memory in the data flow Properties dialog box. • Sorted input: Reads the comparison table in the order of the primary key columns using sequential read. Improves performance because Data Services reads the comparison table only once. To take advantage of this option, the order of the input data set must match the order of all primary key columns in the Table_Comparison transform. If the input data set order matches the order of all primary key columns, drag the primary key columns from the input schema in the Table_Comparison transform into the Input primary key columns box. Using a sequential read, Data Services reads the comparison table in the order of the primary key columns. If you presort the input data, add a query between the source and the Table_Comparison transform. In the Query transform, drag the primary key columns from the Schema In pane into the Order By box. The columns in the Order By box must match the order of the primary key columns that you drag into the Input primary key columns box of the Table_Comparison transform. This set up explicitly adds a sort operation (ORDER BY) to the query input data set to ensure that the input data set order matches the order of the read from the comparison table in the transform.

Option	Description
Detect deleted row(s) from comparison table	<p>Optional. Generates DELETE rows for all rows that are in the comparison table and not in the input set. Assumes that the input set represents the complete data set. By default this option is turned off.</p> <div> <p>i Note</p> <p>The Table_Comparison transform flags rows as DELETE. However, rows actually become DELETE rows only after the data flow completes all other processing.</p> </div> <p>If you select the following options, an additional section appears in the Table_Comparison editor. The section allows you to specify how to handle DELETE rows with duplicate keys:</p> <ul style="list-style-type: none"> • Generated key column • Detect deleted row(s) from comparison table • Row-by-row select • Sorted input comparison method <p>If you choose these options, also choose one of the following for deleted rows with the same key value:</p> <ul style="list-style-type: none"> • Detect all rows: Outputs detected DELETE rows for all rows. • Detect row with largest generated key value: Outputs detected DELETE rows only for the generated key row with the largest value. If you select this option and the Cached comparison table option, Data Services always deletes the row with largest generated key value. <p>When you leave the Generated key column option blank, then the secondary options with the Detect Deleted row(s) from comparison table are not enabled. If Data Services finds duplicate keys in the comparison table, then the DELETE output is corrupted.</p> <div> <p>i Note</p> <p>If you choose the Detect Deleted row(s) from comparison table option, then the performance of the data flow is slower. The comparison methods most affected are the Row-by-row select, followed by Cached compare table, then Sorted input option. For Row-by-row select and Cached compare table, Data Services processes the deleted rows at the end of the data flow. For Sorted input, Data Services processes deleted rows as they are encountered in the data flow.</p> </div> <p>Run as a separate process: Creates a separate data flow process for the Table_Comparison transform.</p>

Option	Description
<i>Generated Key Column</i>	<p>Optional. Provides a method of processing duplicate keys in the comparison table.</p> <div> <p>❖ Example</p> <p>If you have employee data that includes a social security number as a primary key and multiple entries for some of these keys, specify a column in the comparison table with unique keys as the generated key column.</p> </div> <p>A generated key column indicates which row is used in the comparison of a set that contains identical primary keys. Specify an existing column name that does not have duplicate keys. The Generated key column option insures that:</p> <ul style="list-style-type: none"> • For an UPDATE, the output data set contains the largest generated key found for a given primary key. • For an INSERT, the output data set contains a NULL value for the generated key because Data Services typically omits this column from the Table Comparison transform and sets it by the Key_Generation transform later. • If an input column is used as generated key column, its value is preserved. • For a DELETE, the output data set includes all duplicate key rows or just the row with the largest key value. • If there is more than one row in the comparison table with the same primary key value and generated key value, the transform chooses the row to compare. • If the input data set and the comparison table have the column you specified in <i>Generated key column</i>, the transform does not compare the values for this column; it preserves the value. <p>If you do not set this option, your comparison table contains rows with the same primary keys, and the transform chooses which of the rows to compare.</p>
<i>Input contains duplicate keys</i>	<p>Optional. Provides a method of processing duplicate keys in the input data set. If you have more than one row with the same key in the <i>Input Primary Key</i> box, then select this checkbox. Data Services processes all DUPLICATE, INSERT, UPDATE, and DELETE rows. The INSERT, UPDATE, and DELETE rows occur in the same order as they occur in the input table.</p> <p>If you do not set this option, and your input columns have duplicate keys, the transform chooses the rows to compare during data flow processing.</p> <div> <p>i Note</p> <p>Because this option uses additional memory to track rows with duplicate keys, use this option only when necessary.</p> </div>

Option	Description																
<i>Input primary key column(s)</i>	<p>Specify the input data set columns that uniquely identify each row. These columns must be present in the comparison table with the same column names and data types.</p> <p>Drag each column from the <i>Schema In</i> pane into the <i>Input primary key columns</i> box. The transform selects rows from the comparison table that match the values from the primary key columns in the input data set.</p> <p>If values from more than one column are required to uniquely identify each row in the table, add more than one column to the <i>Input primary key columns</i> box.</p> <div> <p>i Note</p> <p>Do not include nested schemas in the <i>Input primary key columns</i> list.</p> </div>																
<i>Table name</i>	<p>The fully qualified name of the source table from which the maximum existing key is determined (key source table). Import this table into the repository before you set this option. <i>Table name</i> is represented as <code>DATASTORE . OWNER . TABLE</code>. If you use Netezza 7.x, enter <i>Table name</i> as <code>DATASTORE . OWNER . SCHEMA . TABLE</code>:</p> <ul style="list-style-type: none"> DATASTORE is the name of the datastore Data Services uses to access the key source table OWNER, if required, depends on the database type. For details, see the following table: <table> <tr> <th>Database type</th><th>Owner value</th></tr> <tr> <td>DB2</td><td>Data source dependent</td></tr> <tr> <td>Informix</td><td>Informix-defined user name</td></tr> <tr> <td>Microsoft SQL Server</td><td>User name</td></tr> <tr> <td>ODBC</td><td>Data source dependent</td></tr> <tr> <td>Oracle</td><td>User name</td></tr> <tr> <td>SAP Sybase SQL Anywhere</td><td>User Name</td></tr> <tr> <td>SAP Sybase</td><td>User name</td></tr> </table>	Database type	Owner value	DB2	Data source dependent	Informix	Informix-defined user name	Microsoft SQL Server	User name	ODBC	Data source dependent	Oracle	User name	SAP Sybase SQL Anywhere	User Name	SAP Sybase	User name
Database type	Owner value																
DB2	Data source dependent																
Informix	Informix-defined user name																
Microsoft SQL Server	User name																
ODBC	Data source dependent																
Oracle	User name																
SAP Sybase SQL Anywhere	User Name																
SAP Sybase	User name																

Option	Description
<i>Filter</i>	<p>Optional. Limits the rows from the comparison table that are considered for comparison against the input data set.</p> <div> <p>⚠ Caution</p> <p>Incorrect filtering can produce false <code>INSERT</code> rows. Construct your filter expression carefully to avoid unexpected results.</p> </div> <p>Valid filter expressions can contain the following elements:</p> <ul style="list-style-type: none"> Columns from the comparison table Constants such as integers, strings, substitution parameters, global variables, and expressions with these combinations Relational operators such as <code>=</code>, <code><</code>, <code>></code>, <code><=</code>, <code>>=</code>, <code>LIKE</code>, <code>IN</code> Logical operators such as <code>AND</code>, <code>OR</code> <div> <p>i Note</p> <p>Do not use Data Services functions, joins, and other SQL statements in filter expressions.</p> </div> <p>Examples of valid filter expressions:</p> <ul style="list-style-type: none"> <code>TC.col1 IN (1, 2, 3)</code> <code>TC.col1 >= 1 AND TC.col1 < 1000</code> <code>TC.col1 = 1 AND TC.col2 = 2</code> <code>col1 = 10 OR col1 = 20</code> <code>TC.col1 = \$V_DEPTNO</code>

Parent topic: [Table_Comparison \[page 422\]](#)

Related Information

[Data outputs \[page 424\]](#)


[INSERT row data output \[page 425\]](#)

[UPDATE row data output \[page 426\]](#)

8.4.11 XML_Pipeline

Use the XML_Pipeline transform to process large XML inputs in small batches.

XML_Pipeline information

Characteristic	Description
	XML_Pipeline icon
Use	<p>With this transform, Data Services does not read the entire XML input into memory and build an internal data structure before performing the transformation. An NRDM structure is not required to represent the entire XML data input. Instead, the XML_Pipeline transform uses a portion of memory to process each instance of a repeatable structure. Then the transform continually releases and reuses memory to steadily flow XML data through the transform.</p> <p>During execution, Data Services pushes operations of the XML_Pipeline transform to the XML source.</p> <p>Limitation: Because the XML_Pipeline transform pushes operations to the XML source during execution, do not use a breakpoint between your XML source and an XML_Pipeline transform.</p>
Data inputs	<p>XML file or XML message.</p> <p>Connect more than one XML_Pipeline transform to an XML source.</p>
Data outputs	N/A

[Rules for setting input and output schemas \[page 434\]](#)

There are specific rules to follow when you drag objects from the *Schema In* to the *Schema out* panes.

[XML_Pipeline editor \[page 434\]](#)

Specify input and output schemas in the XML_Pipeline transform editor.

[XML_Pipeline transform example \[page 435\]](#)

View the example data flow that contains the XML_Pipeline transform to help you understand how to use it in your jobs.

Parent topic: [Data Integrator transforms \[page 357\]](#)

Related Information

[Data_Transfer \[page 358\]](#)

[Date_Generation \[page 372\]](#)

[Effective_Date \[page 375\]](#)

[Hierarchy_Flattening \[page 380\]](#)

[History_Preserving \[page 392\]](#)

[Key_Generation \[page 401\]](#)

[Map_CDC_Operation \[page 405\]](#)

[Pivot \(Columns to Rows\) \[page 413\]](#)

[Reverse Pivot \(Rows to Columns\) \[page 418\]](#)

[Table_Comparison \[page 422\]](#)

8.4.11.1 Rules for setting input and output schemas

There are specific rules to follow when you drag objects from the *Schema In* to the *Schema out* panes.

- Do not drag the root level schema from the input schema to the output schema.
- Drag the same child object as many times as necessary from the input schema to the output schema only when each instance of the object has a unique name. Rename the mapped instance before attempting to drag an object to the output for the second and subsequent time.
- If you drag a column or sub schema from the input to the output schema, do not map the parent schema for that column or sub schema. Similarly, if you drag a parent schema from the input to the output schema, do not map an individual column or sub schema from under that parent.
- Do not map items from two sibling repeating sub schemas. This mapping causes a Cartesian product. The XML_Pipeline transform does not support Cartesian products, which involves combining every row from one table with every row in another table of two repeatable schemas.

Parent topic: [XML_Pipeline \[page 433\]](#)

Related Information

[XML_Pipeline editor \[page 434\]](#)

[XML_Pipeline transform example \[page 435\]](#)

8.4.11.2 XML_Pipeline editor

Specify input and output schemas in the XML_Pipeline transform editor.

In addition to input and output schema panes, the *Mapping* tab displays how Data Services maps any selected output column.

When connected to an XML source, the XML_Pipeline transform editor shows the input and output schema structures as a root schema containing repeating and non-repeating sub-schemas. The Designer shows this information with the icons described in the following table.

	Root schema and repeating sub schema
	Non-repeating sub schema

Within each sub schema, the Designer shows mapped and unmapped columns using the icons described in the following table.

	Column not used in output mapping
	Column used in output mappings

Parent topic: [XML_Pipeline \[page 433\]](#)

Related Information

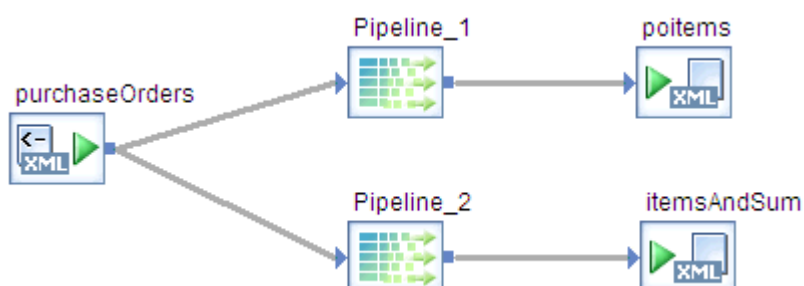
[Rules for setting input and output schemas \[page 434\]](#)

[XML_Pipeline transform example \[page 435\]](#)

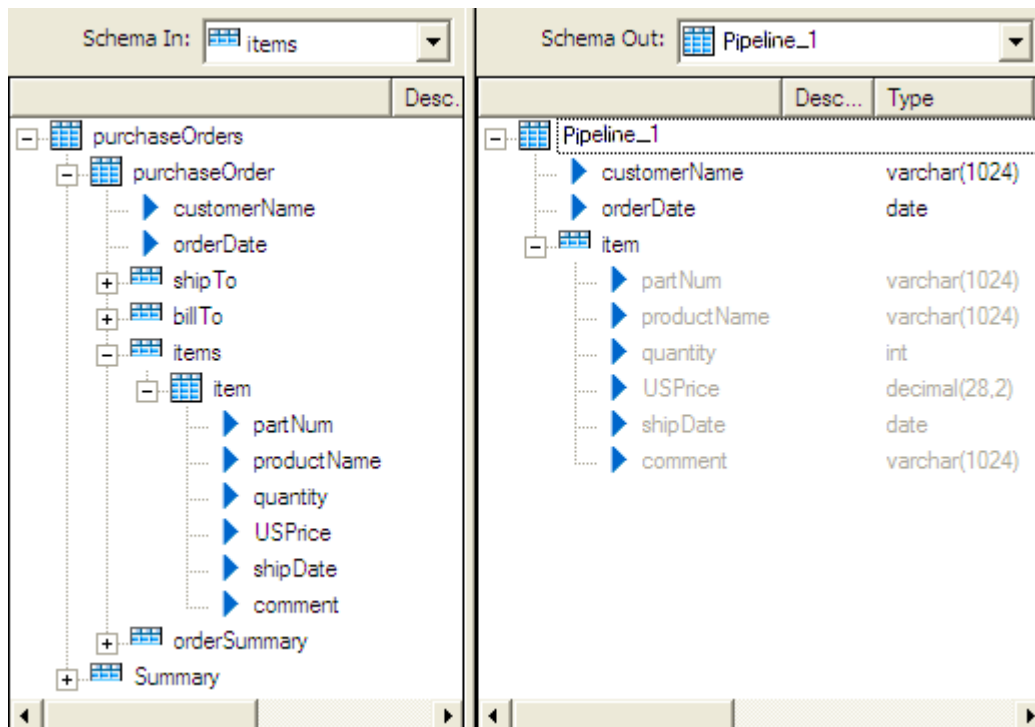
8.4.11.3 XML_Pipeline transform example

View the example data flow that contains the XML_Pipeline transform to help you understand how to use it in your jobs.

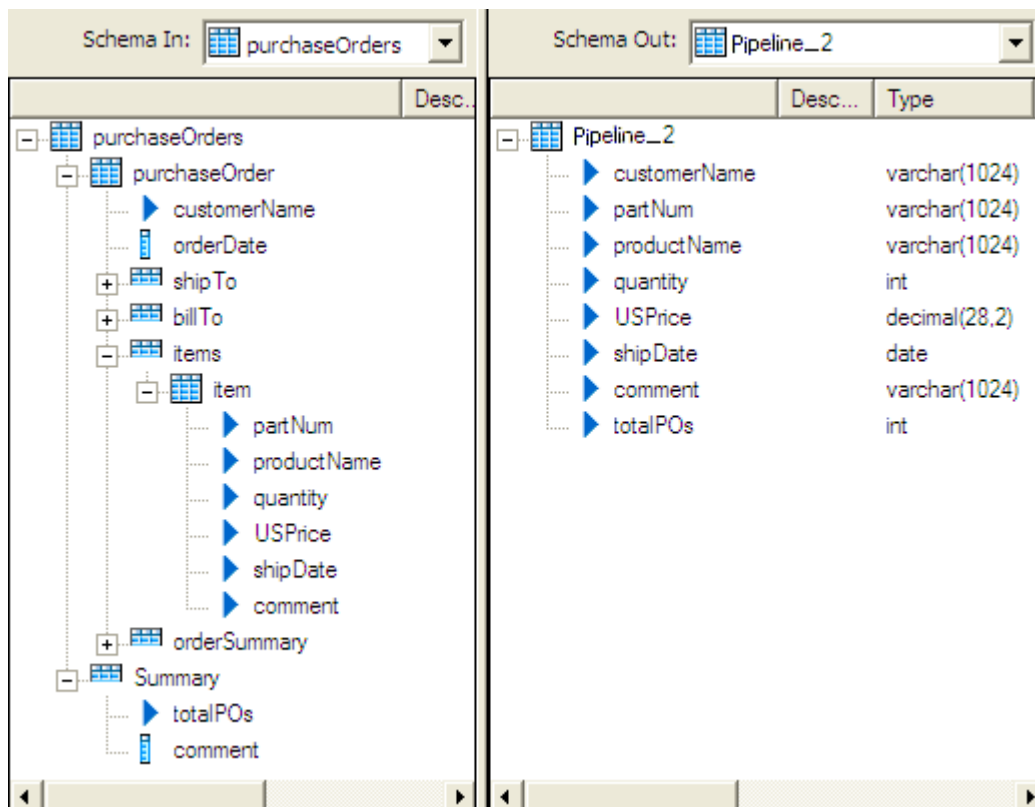
This simple data flow contains two XML_Pipeline transforms, but only one of them takes full advantage of the pipelining power.



- The "XML_Pipeline_1" transform allows XML data to flow because the repeatable column, purchaseOrders.purchaseOrder.items.item is selected. The XML source produces one row after parsing one item.



- The "XML_Pipeline_2" transform does not take advantage of the pipelining power because the purchaseOrders.Summary.totalPOs column is selected and this column occurs structurally after the repeatable column (purchaseOrders.purchaseOrder.items.item). In this scenario, the XML source must assemble the entire structure of items in memory before processing.



However, if you broke this up into two XML_Pipeline transforms, mapping the purchaseOrders.Summary.totalPOs column in a separate transform, you could connect both XML_Pipeline transforms to a Query transform and take advantage of the reduced memory consumption to get the same output.

Parent topic: [XML_Pipeline \[page 433\]](#)

Related Information

[Rules for setting input and output schemas \[page 434\]](#)

[XML_Pipeline editor \[page 434\]](#)

[Query \[page 943\]](#)

8.5 Data Quality transforms

Transforms that help you improve the quality of your data.

The Data Quality transforms can parse, standardize, correct, enrich, match, and consolidate your customer and operational information assets.

[Blueprints and other content objects for download \[page 438\]](#)

Blueprints are partially created jobs that you can use to set up your own jobs for common scenarios.

[About Data Quality fields \[page 445\]](#)

Many transforms require mapped input and generated output fields.

[About data quality statistics \[page 448\]](#)

The data quality statistics tables enable you to analyze many aspects of the quality of your data.

[Associate \[page 454\]](#)

The Associate transform combines or associates matching results generated from the Match transform.

[Country ID \[page 472\]](#)

The Country ID transform parses your input data and then identifies the country of destination for each record.

[Data Cleanse \[page 474\]](#)

Use the Data Cleanse transform to parse and format custom or person and firm data as well as phone numbers, dates, e-mail addresses, and Social Security numbers.

[DSF2® Walk Sequencer \[page 514\]](#)

Use the DSF2 Walk Sequencer transform to add walk sequencing information to your output data.

[Geocoder \[page 524\]](#)

Use the Geocoder transform to append latitude, longitude, and US census data to your data.

[Global Address Cleanse \[page 552\]](#)

Use the Global Address Cleanse transform to identify, parse, validate, and correct your global address data.

[Global Suggestion List \[page 640\]](#)

The Global Suggestion List transform queries addresses with minimal data, and offers suggestions to complete the address.

[Match \[page 651\]](#)

[USA Regulatory Address Cleanse \[page 710\]](#)

[Address Cleanse reference \[page 765\]](#)

[Data Cleanse reference \[page 823\]](#)

8.5.1 Blueprints and other content objects for download

Blueprints are partially created jobs that you can use to set up your own jobs for common scenarios.

When you use blueprints, you save time because we have configured many settings based on the specific scenario. Data Services content objects include custom functions, best practices, the data quality statistics table, and a translation file.

We periodically post new and updated blueprints and other SAP Data Services content and information to the SAP Community Network Web site for your use.

Blueprints

We have identified a number of common scenarios that you are likely to handle with Data Services. Instead of creating your own job from scratch, look through the blueprints. If you find one that is closely related to your particular business problem, use the blueprint by modifying the settings for your specific needs.

For each scenario, we include a blueprint that is already set to solve the business problem in that scenario. Each blueprint contains the necessary project, jobs, data flows, file formats, sample data, template tables, and custom functions. Run the data flows in your environment with only a few modifications.

You can download all of the blueprints or only the blueprints and other content that you find useful from the SAP Community Network Web site. Refer to this site frequently for updated content and use the forums to ask us any questions or to make requests. We have also provided the ability for you to upload and share any content that you have developed with the rest of the development community.

Transform configurations

A transform configuration is a transform with preconfigured input fields, output fields, and options that can be used in multiple data flows. These are useful if you repeatedly use a transform with specific options and input and output fields.


When Data Services is installed, read-only transform configurations are provided for the Data Quality transforms.

You can use transform configurations in your data flows or as an example of a typical transform. After you place an instance of the transform configuration in a data flow, you can override these preset defaults. You can also create your own transform configuration, either by replicating an existing transform configuration or creating a new one.

Task localization file for data quality statistics

The content in the data quality statistics tables need to be translated into recognizable terms and descriptions. We provide the necessary translation file for you to download.

When you use the Data Cleanse, Geocoder, and Global Address Cleanse transforms you can choose to generate the data quality statistics tables. The software sends data quality statistics to the corresponding tables in the repository based on the job set up. The tables contain lines of identification codes generated by the software about the various data quality processes performed on your data. The identification codes require a translation file that converts the identification codes into recognizable terms or descriptions.

The task localization file name is `dqs_task_localization_<language>.csv`. We provide the file in languages that are used most commonly in Data Services. Download it from the SAP Community Network website at: <http://scn.sap.com/docs/DOC-68523> .

Parent topic: [Data Quality transforms \[page 437\]](#)

Related Information

[About Data Quality fields \[page 445\]](#)
[About data quality statistics \[page 448\]](#)
[Associate \[page 454\]](#)
[Country ID \[page 472\]](#)
[Data Cleanse \[page 474\]](#)
[DSF2® Walk Sequencer \[page 514\]](#)
[Geocoder \[page 524\]](#)
[Global Address Cleanse \[page 552\]](#)
[Global Suggestion List \[page 640\]](#)
[Match \[page 651\]](#)
[USA Regulatory Address Cleanse \[page 710\]](#)
[Address Cleanse reference \[page 765\]](#)
[Data Cleanse reference \[page 823\]](#)

8.5.1.1 Transform configurations

Designer provides Data Quality transform configurations that are partially configured for specific types of situations.

The following Data Quality transform configurations are available from the [Transforms](#) tab of the object library. When you use them in your data flows, complete the configuration based on your business case.

Associate transform

Transform configuration	Description
AssociateGroupStatistics_AssociateBatch	A sample Associate transform that generates group statistics.
Base_Associate	A sample base Associate transform that you can use to configure an Associate transform.
Wizard_AssociateBatch	A sample Associate transform configuration that you use with the Match Wizard. Do not edit this configuration.

Country ID transform

Transform configuration	Description
CountryID2Char	A sample Country ID transform that generates the two-character ISO country code.

Data Cleanse transform

Transform configuration	Description
Base_DataCleanse	A sample base Data Cleanse transform that you use to configure a base to a custom Data Cleanse transform.
Chinese_DataCleanse	A sample Data Cleanse transform configured to cleanse name, title, firm, date, e-mail, and phone data using Chinese based data quality rules.

Transform configuration	Description
Dutch_DataCleanse	A sample Data Cleanse transform configured to cleanse name, title, firm, date, e-mail, and phone data using Dutch based data quality rules.
EnglishNorthAmerica_DataCleanse	A sample Data Cleanse transform configured to cleanse name, title, firm, date, e-mail, and phone data using English based data quality rules.
French_DataCleanse	A sample Data Cleanse transform configured to cleanse name, title, firm, date, e-mail, and phone data using French based data quality rules.
German_DataCleanse	A sample Data Cleanse transform configured to cleanse name, title, firm, date, e-mail, and phone data using German based data quality rules.
Italian_DataCleanse	A sample Data Cleanse transform configured to cleanse name, title, firm, date, e-mail, and phone data using Italian based data quality rules.
Japanese_DataCleanse	A sample Data Cleanse transform configured to cleanse name, title, firm, date, e-mail, and phone data using Japanese based data quality rules.
Portuguese_DataCleanse	A sample Data Cleanse transform configured to cleanse name, title, firm, date, e-mail, and phone data using Portuguese based data quality rules.
Spanish_DataCleanse	A sample Data Cleanse transform configured to cleanse name, title, firm, date, e-mail, and phone data using Spanish based data quality rules.

Geocoder transform

Transform configuration	Description
Geocode	A sample Geocoder transform configured to assign latitude and longitude based on an address or point-of-interest reference point.
ResultListGeocode	A sample Geocoder transform configured to provide a list of addresses or points of interest based on an address or latitude and longitude reference point.
ReverseGeocode	A sample Geocoder transform configured to assign an address based on a latitude and longitude reference point.

Global Address Cleanse transform

Transform configuration	Description
Australia_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Australia.
Brazil_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Brazil.
Canada_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Canada.
China_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in China.
Europe_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in multiple European countries.
France_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in France.
Germany_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Germany.

Transform configuration	Description
Global_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse Latin-1 address data in any supported country.
GlobalSuggestions_Address_Cleanse	A sample Global Address Cleanse transform configured to cleanse Latin-1 address data in any supported country using the Suggestion List feature.
Greece_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Greece when the address data consists of Greek data.
Italy_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Italy.
Japan_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Japan when the address data consists of Japanese kanji, katakana, and hiragana.
Portugal_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Portugal.
Spain_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Spain.
UK_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in the United Kingdom.
USA_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in the United States.
USASuggestions_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in the United States using the Suggestion List feature.

Global Suggestion List transform

Transform configuration	Description
GlobalSuggestions	A sample Global Suggestion List transform configured to generate a suggestion list for Latin-1 address data in any supported country.
UKSuggestions	A sample Global Suggestion List transform configured to generate a suggestion list for partial address data in the United Kingdom.

Match transform

Transform configuration	Description
Address_MatchBatch	A sample Match transform configured to identify matching data records based on similar address data.
AddressJapan_MatchBatch	A sample Match transform configured to identify matching data records based on similar address data when the data consists of Japanese kanji, katakana, and hiragana.
AddressSingleField_MatchBatch	A sample Match transform configured to identify matching data records based on similar address data (when the address data is in a single field).
Base_Match	A base Match configuration that is used to configure a Match transform without necessarily performing matching.
ConsumerHouseholdResFamInd_MatchBatch	A sample Match transform configured to identify three levels of matching data records: residence based on similar address data, family based on similar family name data, and individual based on similar given name and postname data.

Transform configuration	Description
ConsumerHouseholdResInd_MatchBatch	A sample Match transform configured to identify two levels of matching data records: residence based on similar address data, and individual based on similar name data.
CorporateHouseholdFirmInd_MatchBatch	A sample Match transform configured to identify two levels of matching data records: firm based on similar firm and address data, and individual based on similar name data.
FirmAddress_MatchBatch	A sample Match transform configured to identify matching data records based on similar firm and address data.
FirmAddressJapan_MatchBatch	A sample Match transform configured to identify matching data records based on similar firm and address data when the data consists of Japanese kanji, katakana, and hiragana.
IndividualId_MatchBatch	A sample Match transform configured to identify matching data records based on the same individual identification number data.
NameAddress_MatchBatch	A sample Match transform configured to identify matching data records based on similar name and address data.
NameDate_MatchBatch	A sample Match transform configured to identify matching data records based on similar name and date data.
NameEmail_MatchBatch	A sample Match transform configured to identify matching data records based on similar name and e-mail data.
NameFirmAddress_MatchBatch	A sample Match transform configured to identify matching data records based on similar name, firm, and address data.
NameFirmAddressJapan_MatchBatch	A sample Match transform configured to identify matching data records based on similar name, firm, and address data when the data consists of Japanese kanji, katakana, and hiragana.
NameIndividualId_MatchBatch	A sample Match transform configured to identify matching data records based on similar name and individual identification number data.
NamePhone_MatchBatch	A sample Match transform configured to identify matching data records based on similar name and phone data.
ProductDescription_MatchBatch	A sample Match transform configured to identify matching data records based on similar product description data.
Wizard_MatchBatch	A sample Match configuration used by the Match Wizard. Do not edit this configuration.

USA Regulatory Address Cleanse transform


Transform configuration	Description
USARegulatory_AddressCleanse	A sample USA Regulatory Address Cleanse transform configured to cleanse address data according to CASS requirements.
USARegulatoryEWS_AddressCleanse	A sample USA Regulatory Address Cleanse transform configured to cleanse address data according to CASS requirements, with Early Warning System.
USARegulatoryGeo_AddressCleanse	A sample USA Regulatory Address Cleanse transform configured to cleanse address data according to CASS requirements, with GeoCensus.
USARegulatoryNCOALink_AddressCleanse	A sample USA Regulatory Address Cleanse transform configured to cleanse address data using NCOALink data.

Transform configuration	Description
USARegulatoryNonCertified_ AddressCleanse	A sample USA Regulatory Address Cleanse transform configured to cleanse address data in non certified mode.
USARegulatoryNonCertifiedGeo_ AddressCleanse	A sample USA Regulatory Address Cleanse transform configured to cleanse address data in non certified mode, with GeoCensus.
USARegulatoryRDI_ AddressCleanse	A sample USA Regulatory Address Cleanse transform configured to cleanse address data according to CASS requirements, with Residential Delivery Indicator.
USARegulatorySuggestions_ AddressCleanse	A sample USA Regulatory Address Cleanse transform configured to cleanse address data using the Suggestion List feature.
USARegulatoryZ4Change_ AddressCleanse	A sample USA Regulatory Address Cleanse transform configured to cleanse address data according to CASS requirements, with Z4Change.
DSF2_Walk_Sequencer	A sample DSF2 Walk Sequencer transform configured to append sequence information to your data so that, by using presort software, you can obtain walk sequence discounts.

8.5.1.2 Downloading the data quality statistics task localization file

When you query the data quality statistics files, you need the task localization file to translate the codes from the tables into understandable and useful information.

To download the SAP Data Services task localization file:

1. Open the Data Services SAP Community Network at <http://scn.sap.com/docs/DOC-68523> .
2. Find the applicable task localization file based on the language you need, and what is available. The file name is `dqs_task_localization_<language>.csv`.
3. Follow the download instructions in the SCN Data Quality Statistics Task Localization page.

Related Information

[Blueprints and other content objects for download \[page 438\]](#)

8.5.1.3 Downloading blueprints and other content objects

Learn how to download and install blueprints and other content objects.

Go to the SAP Community Network at <https://wiki.scn.sap.com/wiki/display/EIM/SAP+Data+Services+Blueprints> .

1. Open the *Content Objects User Guide* to view a list of all of the available blueprints and content objects and their descriptions.

The *Content Objects User Guide* also contains instructions for downloading and setting up the blueprints.

2. Select the blueprint that you want to download.
3. Follow the instructions in the *Content Objects User Guide* to download the files to the appropriate location and make the necessary modifications in SAP Data Services to run the blueprints.

8.5.2 About Data Quality fields

Many transforms require mapped input and generated output fields.

Map these fields in the [Input](#) and [Output](#) tabs of the transform editor. For detailed information about the input and output fields applicable for each transform, see the individual transform topics.

Parent topic: [Data Quality transforms \[page 437\]](#)

Related Information

[Blueprints and other content objects for download \[page 438\]](#)

[About data quality statistics \[page 448\]](#)

[Associate \[page 454\]](#)

[Country ID \[page 472\]](#)

[Data Cleanse \[page 474\]](#)

[DSF2® Walk Sequencer \[page 514\]](#)

[Geocoder \[page 524\]](#)

[Global Address Cleanse \[page 552\]](#)

[Global Suggestion List \[page 640\]](#)

[Match \[page 651\]](#)

[USA Regulatory Address Cleanse \[page 710\]](#)

[Address Cleanse reference \[page 765\]](#)

[Data Cleanse reference \[page 823\]](#)

8.5.2.1 Content types

All input and output fields have an assigned content type that identifies the type of data in the field.

Setting the content type helps you map your fields when you set downstream transforms. The software searches all upstream fields and automatically maps the fields that have a content type that is relevant to the type of transform that you're currently mapping. Those upstream fields will automatically be added as mapped fields.

❖ Example

You have a column of data in your data source that is comprised mostly of first names called Given Name1. The software automatically maps the field to a Data Quality-recognized content type, Given_Name1.

You can change the automatic content type in the *Column Properties* window. Open the window by double-clicking the field name in the *Schema Out* grid of the transform editor. Select a different content type from the *Content Type* drop list.

You can specify the content type directly in your source data in the object Library, including source data from XML Schemas, COBOL Copybooks, flat files, Excel, and IDOC objects. If the data in a column cannot be automatically mapped, then you will see a blank content type. Content types in an XML Schema can be changed in the Designer object library, and the change is reflected in all data flows where the schema is used. You cannot change the content type in an XML Schema from within a single data flow.

A few situations to be aware of

If you are importing an XSD or DTD format, you can have the software automatically assign the content by selecting *Automatically Assign Content Type* in the *Import DTD Format* window.

If you attempt to merge the contents of two corresponding columns in the Merge transform, be certain that the content types match. If the content types do not match, the software issues a warning message when you validate.

If you have an input source on your local repository and overwrite it with one from the central repository, then your content type information is overwritten.

If you reimport a column or table in Designer, the content types for all of the existing columns are preserved by default. For instructions to change the default setting for content type, see “Editing default content type preserve setting”.

Available content types

The following content types are available for each transform.

- <Blank>
- Address
- Address Primary Name
- Address Primary Number
- Address Primary Postfix
- Address Primary Prefix
- Address Primary Type
- Address Secondary Number
- Country
- Date
- Delivery Point

- DPV Status
- Email
- Family Name1
- Family Name1 Match Std
- Family Name2
- Family Name2 Match Std
- Firm
- Firm Location
- Firm Location Match Std
- Firm Match Std
- Given Name1
- Given Name1 Match Std
- Given Name2
- Given Name2 Match Std
- Group Number
- Locality
- Lot
- Lot Order
- Name
- Name And Firms
- Phone
- Postcode
- Postcode1
- Postcode2
- Postname
- Postname Match Std
- Prenom
- Prenom Match Std
- Region
- Sortcode Rte
- SSN
- Title
- Title Match Std

Related Information

[Editing default content type preserve setting \[page 448\]](#)

8.5.2.1.1 Editing default content type preserve setting

SAP Data Services has default settings for field attribute values that controls how the software preserves data when you reimport table and columns.

To change the default behavior when you re import a column and you want the software to clear existing attribute values:

1. In *Designer*, select **Tools > Options > .**
2. In *Options*, expand *Designer* and select *Attribute Values*.
3. Select either *Column* or *Table* from the *Object Type* dropdown list.
4. Make default settings for each applicable attribute.

❖ Example

To change the default setting for *Content Type*, click the corresponding *Action* dropdown list and select *Clear*. When you reimport a column, the software clears existing content type attributes.

8.5.3 About data quality statistics

The data quality statistics tables enable you to analyze many aspects of the quality of your data.

The data quality statistics tables in the repository contain summary and record-level statistics about the cleansing and assignment processes performed on your data. You can generate the tables by setting the Data Quality Statistics options in the applicable transforms: Data Cleanse, Geocoder, Global Address Cleanse.

Parent topic: [Data Quality transforms \[page 437\]](#)

Related Information

[Blueprints and other content objects for download \[page 438\]](#)

[About Data Quality fields \[page 445\]](#)

[Associate \[page 454\]](#)

[Country ID \[page 472\]](#)

[Data Cleanse \[page 474\]](#)

[DSF2® Walk Sequencer \[page 514\]](#)

[Geocoder \[page 524\]](#)

[Global Address Cleanse \[page 552\]](#)

[Global Suggestion List \[page 640\]](#)

[Match \[page 651\]](#)

[USA Regulatory Address Cleanse \[page 710\]](#)

[Address Cleanse reference \[page 765\]](#)

[Data Cleanse reference \[page 823\]](#)

[Data quality statistics common requirements \[page 449\]](#)
[Data quality statistics transform set up requirements \[page 451\]](#)

8.5.3.1 Data quality statistics common requirements

The following table contains requirements for data quality statistics that apply to more than one statistics table type.

Requirement	Statistics table
<p>Require the task localization file. This file converts raw data from the data quality statistics tables into understandable language when you extract statistics.</p> <p>The task localization file is available to download from our SCN.</p>	All data quality statistics tables
<p>Do not require the Row_ID field in the output field set up. These statistic tables contain summary statistics and the Row_ID field is required only for non-summary statistics tables.</p> <p>The software issues a warning when your job set up:</p> <ul style="list-style-type: none"> Includes the Row_ID output field and is not set up to generate non-summary data quality statistics. Does not include the Row_ID output field and is set up to generate non-summary data quality statistics. 	<p>CLEANSE_STATISTICS_ GEOCODE_STATISTICS_</p>
<p>Require the Row_ID field in the output field set up. These are non-summary statistic tables and the Row_ID field is required to join the transform input or output with the output statistics.</p> <p>The software issues a warning when your job set up:</p> <ul style="list-style-type: none"> Does not include Row_ID output field and is set up to generate non-summary data quality statistics. Includes the Row ID output field and is not set up to generate non-summary data quality statistics. <p>You can use Row_ID to join the transform input or output with the non-summary statistics table information. For example, use Row_ID to drill into the before-and-after changes in a specific component that was cleansed.</p>	<p>CLEANSE_INFO_CODES_ CLEANSE_CHANGE_INFO_ CLEANSE_COMPONENT_INFO_ (available in a future release, after version 14.2.6.0) CLEANSE_ADDRESS_RECORD_INFO_ GEOCODE_INFO_CODES_</p>

Requirement	Statistics table
Require that you map an input file primary key to the output file. You can then use the mapped primary key to join the input with any of the data quality statistics reports.	CLEANSE_INFO_CODES_ CLEANSE_CHANGE_INFO_ CLEANSE_COMPONENT_INFO_ (available in a future release, after version 14.2.6.0) CLEANSE_ADDRESS_RECORD_INFO_ GEOCODE_INFO_CODES_
<p>Require that you set up a field for data cleansing when you want the software to generate data cleansing statistics for that field. If you do not set up a field for data cleansing, the software does not generate data cleansing statistics for that field.</p> <p>For example, an input file contains firm, address, and phone information. You set up the firm and address fields for data cleansing, but you do not set up the phone field for data cleansing. The software generates statistics that include the firm name and address information, but not the phone information.</p>	Applies to all data quality statistics tables.
Can be generated by both Data Cleanse and Global Address Cleanse transforms.	<ul style="list-style-type: none"> • CLEANSE_INFO_CODES_ • CLEANSE_CHANGE_INFO_ • CLEANSE_STATISTICS_ • CLEANSE_COMPONENT_INFO_ (available in a future release, after version 14.2.6.0)

Related Information

[Blueprints and other content objects for download \[page 438\]](#)

[Report and Analysis \[page 478\]](#)

[Data quality statistics tables and supplemental content information \[page 1344\]](#)

[Data quality statistics transform set up requirements \[page 451\]](#)

8.5.3.2 Data quality statistics transform set up requirements

To successfully generate the data quality statistics tables for Data Cleanse, Geocoder, and Global Address Cleanse, you must adhere to specific job set up specifications or the software will not generate data quality statistics.

Related Information

[Data Cleanse input field requirements for data quality statistics \[page 451\]](#)

[Data Cleanse requirements for generating CLEANSE_COMPONENT_INFO statistics table \[page 452\]](#)

[Geocoder job set up requirements for data quality statistics \[page 453\]](#)

[Global Address Cleanse requirements for generating CLEANSE_COMPONENT_INFO table \[page 454\]](#)

8.5.3.2.1 Data Cleanse input field requirements for data quality statistics

Supported and unsupported input fields and input field configurations for generating data quality statistics tables in the Data Cleanse transform.

The following table lists the supported and unsupported input fields for generating data quality statistics in the Data Cleanse transform. It also includes specific input field configurations that are not supported for generating data quality statistics.

Supported and unsupported input fields and configurations

Supported input fields	Unsupported input fields	Unsupported input field configurations
<ul style="list-style-type: none">• EMAIL#• FIRM_LINE#• Discrete firm• NAME_LINE#• NAME_OR_FIRM_LINE#• Discrete Person• PHONE#• TITLE_LINE#	<ul style="list-style-type: none">• DATE#• MULTILINE#• SSN#• UDPM#	<p>The software issues a warning when any of the following configurations exist in the transform with data quality statistics enabled:</p> <ul style="list-style-type: none">• Multiline input fields• Name/title associated fields• Discrete person input fields and NAME_LINE input fields mapped at the same time• Discrete firm input fields and FIRM_LINE input fields mapped at the same time

Related Information

[Data Cleanse requirements for generating CLEANSE_COMPONENT_INFO statistics table \[page 452\]](#)

8.5.3.2.2 Data Cleanse requirements for generating CLEANSE_COMPONENT_INFO statistics table

When you generate the CLEANSE_COMPONENT_INFO_ table in the Data Cleanse or Global Address Cleanse transforms (available after Data Services version 14.2.6.0), there are output field configuration requirements for parsing person, firm, phone, and email fields.

The CLEANSE_COMPONENT_INFO statistics table contains the exact output field location (start position and length) of individual data elements that were parsed during the cleansing process. You can choose to generate this table in either the Data Cleanse transform or the Global Address Cleanse transform.

Person parsing

The software generates output field position data only for Person output fields that have the following characteristics:

- PARENT_COMPONENT = DUAL_NAME* or PERSON*
- FIELD_CLASS = STANDARDIZED

The software does not generate output field position data for Person output fields that have the following FIELD_NAME values (exceptions to the above-supported person field characteristics):

- NAME_SPECIAL
- GENDER
- GENDER_ID
- RULE_LABEL
- SCORE
- *_MATCH_STD*
- MATCH_*

Firm parsing

The software generates output field position data for Firm output fields that have the following characteristics:

- PARENT_COMPONENT = FIRM*
- FIELD_CLASS = STANDARDIZED

The software does not generate output field position data for Firm output fields that have the following FIELD_NAME values (exceptions to the above-supported firm field characteristics):

- RULE_LABEL
- SCORE
- *_MATCH_STD*
- MATCH_*

Phone parsing

The software generates output field position data for Phone output fields that have the following characteristics:

- PARENT_COMPONENT = PHONE*
- FIELD_CLASS = STANDARDIZED

The software does not generate output field position data for Phone output fields that have the following FIELD_NAME values (exceptions to the above-supported phone field characteristics):

- *_MATCH_STD*
- MATCH_*

Email parsing

The software generates output field position data for Email output fields that have the following characteristics:

- PARENT_COMPONENT = EMAIL*
- FIELD_NAME = EMAIL
- FIELD_CLASS = STANDARDIZED

8.5.3.2.3 Geocoder job set up requirements for data quality statistics

There are requirements when you set up the Geocoder transform for generating data quality statistics.

Applicable Geocoder modes

Supported mode	Unsupported mode
Address geocoding mode	Reverse geocoding mode <ul style="list-style-type: none"> • Reverse geocoding mode • Point of interest (POI) textual search mode <p>The software issues a warning when you are in these modes and you select to generate data quality statistics.</p>

See the Geocoder section of the *Designer Guide* for information about Geocoder, Geocoder processing modes, and job set up requirements for each mode.

8.5.3.2.4 Global Address Cleanse requirements for generating CLEANSE_COMPONENT_INFO table

The Global Address Cleanse transform generates data quality statistics for applicable fields when you execute your jobs. However, there are restrictions for your job set up if you choose to generate the CLEANSE_COMPONENT_INFO_ table (available after version 14.2.6.0).


When you enable the CLEANSE_COMPONENT_INFO statistics table, the following job set up restrictions apply:

- Input field position data (start position and length) is not generated for NW_ input fields
- Output field position data (start position and length) is generated for output fields with the following characteristics: FIELD_CLASS = BEST, FIELD_CATEGORY = COMPONENT, and FIELD_ADDRCLASS = DELIVERY or DUAL.
- Output field position data (start position and length), is not generated for the following output fields (these are exceptions to the above supported field characteristics): Multiline1-12 fields and MATCH_ fields

8.5.4 Associate

The Associate transform combines or associates matching results generated from the Match transform.

Associate information

Characteristic	Description
	Associate icon
Use	<p>The Associate transform works downstream from the Match transform to provide a way to combine or associate match results by using the group number generated by the Match transform.</p> <p>Optionally add a Group Statistics operation to the Associate transform to gather match statistics. Combine the results of two or more Match transforms, two or more Associate transforms, or any combination of the two.</p> <p>For example, you may use one Match transform to match on name and address, use a second Match transform to match on SSN, and then use an Associate transform to combine the match groups produced by the two Match transforms.</p>
Content objects	Transform configurations, blueprints, and other content objects.

[Associate transform options \[page 455\]](#)

Transform level options for the Associate transform appear in the [Association Editor](#), which you open using the [Tools](#) menu.

[Association options \[page 456\]](#)

Association options for the Associate transform appear in the [Association Editor](#), which you open using the [Tools](#) menu.

[Post-association processing \[page 457\]](#)

The [Association Editor](#) contains a post processing page, which you use to set up any post processing operations.

[Output fields for the Associate transform \[page 471\]](#)

The Associate transform requires that you map one field on output: Group_Number.

Parent topic: [Data Quality transforms \[page 437\]](#)

Related Information

[Blueprints and other content objects for download \[page 438\]](#)

[About Data Quality fields \[page 445\]](#)

[About data quality statistics \[page 448\]](#)

[Country ID \[page 472\]](#)

[Data Cleanse \[page 474\]](#)

[DSF2® Walk Sequencer \[page 514\]](#)

[Geocoder \[page 524\]](#)

[Global Address Cleanse \[page 552\]](#)

[Global Suggestion List \[page 640\]](#)

[Match \[page 651\]](#)

[USA Regulatory Address Cleanse \[page 710\]](#)

[Address Cleanse reference \[page 765\]](#)




[Data Cleanse reference \[page 823\]](#)

[Transform configurations \[page 440\]](#)

[Blueprints and other content objects for download \[page 438\]](#)

8.5.4.1 Associate transform options

Transform level options for the Associate transform appear in the [Association Editor](#), which you open using the [Tools](#) menu.

To open the [Association Editor](#), select the Associate transform in your data flow and select  [Tools](#)  [Associate Editor](#).  The following table contains descriptions for the options that appear when you click [Transform Options](#) in the left pane of the [Association Editor](#).

Transform Options descriptions

Option	Description
<i>Associate set name</i>	Required. Type a name for your Associate set.
<i>Generate report data</i>	Required. Generate report data for this transform. <i>Yes:</i> The default setting. Generates report data for this transform. <i>No:</i> Turns off report data generation. You would disable report data generation if you do not need to generate reports. For example, if you work in production, you may not need report data. Turning off report data generation improves performance.
<i>Logical source field</i>	Optional. Select the field that contains the ID for the logical source from the dropdown list. The logical source is a group of records that span multiple input sources or a subset of records from a single input source.
<i>Physical source field</i>	Optional. Select the field that contains the ID for the physical source (reader) from the dropdown list.
<i>Run as a separate process</i>	Specify whether to split the transform into a separate process. Running as a separate process can improve performance and throughput by using separate memory and computer resources. <i>Yes:</i> Splits the transform into a separate process. <i>No:</i> The default setting. Keeps the transform in the same process as the rest of the data flow.

Parent topic: [Associate \[page 454\]](#)

Related Information

[Association options \[page 456\]](#)

[Post-association processing \[page 457\]](#)

[Output fields for the Associate transform \[page 471\]](#)

8.5.4.2 Association options

Association options for the Associate transform appear in the *Association Editor*, which you open using the *Tools* menu.

Find the options by selecting the Associate transform in your data flow and selecting **Tools** > *Associate Editor*. The following table contains descriptions for the options that appear when you click *Association<n>* in the left pane of the *Association Editor*. *<n>* represents the association group number. If you have one group, the option is *Association1*.

Add as many association groups as you want.

Associationn option descriptions

Option	Description
Association name	Type a name for this association. Ensure that the name is unique among other associations in this transform.
Group Number field	Click Add Row to add a line to the Group numbers table. Select the field that contains the group number data from the dropdown list. The field is from the Match transform. Click Add Row to add as many group number fields as desired.

Parent topic: [Associate \[page 454\]](#)

Related Information

[Associate transform options \[page 455\]](#)

[Post-association processing \[page 457\]](#)

[Output fields for the Associate transform \[page 471\]](#)

8.5.4.3 Post-association processing

The [Association Editor](#) contains a post processing page, which you use to set up any post processing operations.

Access the [Post associate](#) options by selecting the transform icon in the data flow and selecting ► [Tools](#) ► [Association Editor](#). ►

Add operations to the table by clicking [Add Operation](#). Select one of the following operations from the [Add Operation](#) dropdown list.

- [Group Prioritization](#)
- [Group Statistics](#)
- [Best Record](#)
- [Unique ID](#)

Use the Post associate operations table to navigate to your operations by double-clicking the desired row in the table.

Group Prioritization

Use the Group prioritization operation to order records for processing by other post-match operations. For option descriptions, see [Group prioritization options: Priority Order tab \[page 465\]](#) and [Group prioritization options: Record Completeness tab \[page 466\]](#).

Group statistics

Use Group statistics to generate statistical information about your group of matching records. Statistics include the following information:

- Number of records within the match group
- Sequential group order number
- Group rank

Note

Group rank flags one record within each group of matching records as the master record. It flags all other records in the group as subordinate records.

- Records in a match group that belong to more than one source

Group statistics are essential for generating data for reports. For option descriptions, see [Group Statistics Editor options \[page 463\]](#).

Best record

Use the Best record operation to salvage data from matching records that belong to match groups, and consolidate or post the data to a best record or to all matching records. For option descriptions, see [Best record options: Best Record tab \[page 685\]](#) and [Best record options: Destination Protection tab \[page 462\]](#).

Unique ID

Use the Unique ID operation to have the software assign sequential identification numbers to each new record added to a data warehouse. The software carries the largest sequential number assigned in a project over to the next project and adds 1 to continue the sequential numbering. The new number is the beginning identification number for assigning the next group of sequential identification numbers. occurs when the software processes the next source against the data warehouse file.

Example

The largest sequential number from project 1 is 155. The software starts the numbering for project 2 at 156 for the first unique ID.

For option descriptions, see [Unique ID options: Unique ID tab \[page 467\]](#) and [Unique ID options: Destination Protection tab \[page 470\]](#).

Output record

Use the Output record options to flag certain types of records for potential processing downstream.

[Best record options: Best Record tab \[page 459\]](#)

Use the best record post-match processing operation to update your records with information from other records in a match group, among other things.

[Best record options: Destination Protection tab \[page 462\]](#)

Protect data from being changed by enabling and defining destination protection.

[Group Statistics Editor options \[page 463\]](#)

Use the options in the *Group Statistics Editor* to set up an operation for post association processing.

[Group prioritization options: Priority Order tab \[page 465\]](#)

These options appear in both the *Match Editor* and the *Association Editor* based on which transform you are working with.

[Group prioritization options: Record Completeness tab \[page 466\]](#)

Make settings to set the priority of output data based on record completeness.

[Unique ID options: Unique ID tab \[page 467\]](#)

Use the Unique ID options to assign sequential identification numbers to each new record when adding records to a data warehouse.

[Unique ID options: Destination Protection tab \[page 470\]](#)

Use the Destination Protection tab to control whether a record unique ID is protected based on the source that the record belongs to.

Parent topic: [Associate \[page 454\]](#)

Related Information

[Associate transform options \[page 455\]](#)

[Association options \[page 456\]](#)

[Output fields for the Associate transform \[page 471\]](#)

[Best record options: Best Record tab \[page 685\]](#)

[Group Statistics Editor options \[page 463\]](#)

[Group prioritization options: Priority Order tab \[page 465\]](#)

[Unique ID options: Unique ID tab \[page 467\]](#)

[Output flag selection options \[page 696\]](#)

8.5.4.3.1 Best record options: Best Record tab

Use the best record post-match processing operation to update your records with information from other records in a match group, among other things.

These options appear in both the *Match Editor* and the *Association Editor* based on which transform you are working with.

In the *Post Association Processing* or *Post Matching Processing* dialog of the `<transform_name> Editor`, select to add a Best Record operation. The following options are in the *Best Record Editor* page and the *Best Record* tab.

Best Record post processing operation option descriptions

Option	Description
<i>Best record name</i>	Enter a name for your Best Record operation. Make sure that this name is unique within this Match transform.
<i>Best record strategy</i>	<p>Choose the strategy to determine whether any action is taken on records in a match group. This is the criteria for further action. After you choose the strategy, priority, and field that you want to work with, the Match transform automatically generates the Python code for you (except for Custom).</p> <p><i>Custom</i>: Choose this strategy to base your strategy entirely on custom Python code. This allows you to open the Python Expression editor and create custom Python code.</p> <p><i>Date</i>: Choose Date to base your strategy on a date field.</p> <p><i>Length</i>: Choose Length to base your strategy on the length of data in a field.</p> <p><i>Non_Blank</i>: Choose Non_Blank to base your strategy on the completeness of data in a field.</p> <p><i>Priority_Number</i>: Choose Priority_Number to base your strategy on a number.</p> <p><i>Priority_String</i>: Choose Priority_String to base your strategy on string data in a field.</p>
<i>Strategy priority</i>	<p>These are the choices for priorities for each of the best record strategies, other than Non_Blank and Custom.</p> <p>Date</p> <ul style="list-style-type: none"> <i>Newest</i>: The newest date in the field will cause an action to take place. <i>Oldest</i>: The oldest date in a field will cause an action to take place. <p>Length</p> <ul style="list-style-type: none"> <i>Longest</i>: The longest string in a field will cause an action to take place. <i>Shortest</i>: The shortest string in a field will cause an action to take place. <p>Priority_Number</p> <ul style="list-style-type: none"> <i>Highest</i>: The highest number in a field will cause an action to take place. <i>Lowest</i>: The lowest number in a field will cause an action to take place. <p>Priority_String</p> <ul style="list-style-type: none"> <i>Ascending</i>: The string with the most ascending string order will cause an action to take place. <i>Descending</i>: The string with the most descending string order will cause an action to take place.
<i>Strategy field</i>	Choose a field that contains data that you need to execute your strategy.

Option	Description
<i>Posting destination</i>	<p>Specifies the destination record.</p> <p><i>Master</i>: Post only to a master record.</p> <p><i>Subs</i>: Post only to subordinate records.</p> <p><i>Master to Subs</i>: Push information from the master record and post it to each subordinate record.</p> <p><i>All</i>: Post to both the master and subordinate records.</p>
<i>Post only once per destination</i>	<p><i>Yes</i>: Post only once per destination record. After data is posted to the destination record, the operation stops.</p> <p><i>No</i>: Post more than once per destination. After data is posted to the destination record, the operation continues and the destination record is populated again with the next value. This option should be used when accumulating values such as total sales.</p> <p>Set this option to Yes when you are using the NON_BLANK strategy.</p> <p>Set this option to No when you are using the Longest, Shortest, Newest, Oldest, Ascending, or Descending priorities.</p> <div> <p>Note</p> <p>This option is ignored when using the <i>Master to Subs</i> posting destination. With this posting destination, information can be posted only once.</p> </div>
<i>View/Edit Python</i>	<p>The View/Edit Python button opens the Python Expression editor. If you chose the Custom strategy, you can create your custom Python code. If you chose any other strategy, Python viewed in the editor is read-only.</p>

Best record action fields table

Use the *Best record action fields* table to define the actions taken on the fields based on your strategy.

Option/Option group	Description
<i>Source field</i>	Specifies the name of the source field in the input record.
<i>Destination field</i>	<p>Specifies the name of the destination field in the output record, or the destination of this best record action.</p> <p>You can have the action post to the same input field, or you can post to a different field.</p>
<i>Custom</i>	<p><i>Yes</i>: Specifies that you want to create custom Python code to perform an action on the destination field.</p> <p><i>No</i>: Specifies that you want to use the same source and destination fields.</p> <p>When this option is set to No, the contents of the source field are copied to the destination field.</p>

Option/Option group	Description
<i>Editor</i>	If you chose Yes in the Custom column, a button appears here to allow you open the Python Expression editor and configure your Python code. You can open the Python Expression editor only if Custom is set to Yes.

Parent topic: [Post-association processing \[page 457\]](#)

Related Information

Best record options: [Destination Protection tab \[page 462\]](#)

Group Statistics Editor options [\[page 463\]](#)

Group prioritization options: [Priority Order tab \[page 465\]](#)

Group prioritization options: [Record Completeness tab \[page 466\]](#)

Unique ID options: [Unique ID tab \[page 467\]](#)

Unique ID options: [Destination Protection tab \[page 470\]](#)

8.5.4.3.2 Best record options: Destination Protection tab

Protect data from being changed by enabling and defining destination protection.

Option	Description
<i>Best record name</i>	Enter a name for your Best Record operation. Make sure that this name is unique within this Match transform.
<i>Enable destination protection</i>	Select to protect records from best record operations that may modify the contents.
<i>Default destination protection</i>	Select the default destination protection setting. This is useful because the default setting accounts for records that are protected (or not protected) through the use of sources or fields.
<i>Specify destination protection by field</i>	Select to enable destination protection through a value in a field.
<i>Destination protection field</i>	Choose the field that holds the destination protection value. The field must contain a value of Y or N. Any other value, including blank, causes the default destination protection specification to occur, if you specified a default destination protection.
<i>Specify destination protection by source</i>	Select this option to control destination protection through membership in a particular source. Fill in the table with source names and whether they are protected.
<i>Source Name</i>	Choose the name of the source from the dropdown list. The list here is populated with defined sources and source groups from the Input Sources Editor window of the Match Editor.
<i>Destination protected</i>	Select a value to assign to the source. Select Yes to enable destination for that source. Select No, if you do not wish to protect records from that source.

Parent topic: [Post-association processing \[page 457\]](#)

Related Information

Best record options: [Best Record tab \[page 459\]](#)

Group Statistics Editor options [\[page 463\]](#)

Group prioritization options: [Priority Order tab \[page 465\]](#)




Group prioritization options: [Record Completeness tab \[page 466\]](#)

Unique ID options: [Unique ID tab \[page 467\]](#)

Unique ID options: [Destination Protection tab \[page 470\]](#)

8.5.4.3.3 Group Statistics Editor options

Use the options in the *Group Statistics Editor* to set up an operation for post association processing.

Access the *Group Statistics Editor* by selecting the Associate transform in your data flow and selecting  [Tools](#)  [Associate Editor](#).  Then select Post Association Processing from the left pane and select [Add Operation](#) at right.

Option	Description
<i>Group statistics name</i>	Choose a name for this group statistics operation. If you are including more than one group statistics operation in this Match transform, make sure that the name is unique.
<i>Generate only basic statistics</i>	Select if you want to generate match group statistics. Basic statistics does not include any statistics about input sources.
<i>Generate source statistics from input sources</i>	Select to generate statistic counts about your input sources. You must have input sources defined in the Match editor for this option to be active. If you do not check this option, the Match transform still generates statistics about your match groups.
<i>Generate source statistics from source values</i>	Select to generate source statistics based on source values in a field. If you have a source value field, using this option, you can choose to count all sources or specific ones based on a particular value. When you select this option, the following options become editable: <ul style="list-style-type: none">• Logical source field• Default logical source value• Count all sources• Choose sources to count
<i>Logical source field</i>	Specifies the field that holds the value for your logical sources.

Option	Description
<i>Default logical source value</i>	Specifies a value to use if the field in the Logical source field option is blank. For example, if a record has a blank value in a field, the transform uses the value in the Default logical source value option.
<i>Count all sources</i>	Select to count all sources, no matter what the value in the Logical source field is.
<i>Choose sources to count</i>	<p>Select to specify particular sources to count, based on values in the Logical source field.</p> <p>When you select this option, the following options become editable:</p> <ul style="list-style-type: none"> • Default count flag • Auto generate sources • Predefined count flat field (editable only when you select Auto generate sources) • Manually define logical source count flags
<i>Default count flag value</i>	<p>Specifies the value to use when the Predefined count flag field is invalid. For example, if the Predefined count flag field has data other than Y or N, or it is empty.</p> <p>Yes: Counts all of the records in the source.</p> <p>No: Does not count any of the records in the source.</p>
<i>Auto generate sources</i>	Select to generate sources based on the value in a field.
<i>Predefined count flag field</i>	Specifies the field name that contains the indicator value (Y or N) to determine whether a source is counted. This field is editable only when you also select Auto generate sources .
<p><i>Manually define logical source count flags</i></p> <p>Be sure to fill in both columns for this option to work.</p>	
<i>Source value</i>	Specifies the value in the field to find. This value is case sensitive.
<i>Count</i>	<p>Specifies whether you want to use the value you entered in the Logical source value option in the count.</p> <p>Yes: Includes the logical source value in the count.</p> <p>No: Does not include the value in the logical source value option in the count.</p>

Parent topic: [Post-association processing \[page 457\]](#)

Related Information

Best record options: [Best Record tab \[page 459\]](#)

Best record options: [Destination Protection tab \[page 462\]](#)

Group prioritization options: [Priority Order tab \[page 465\]](#)

Group prioritization options: [Record Completeness tab \[page 466\]](#)

Unique ID options: [Unique ID tab \[page 467\]](#)

Unique ID options: [Destination Protection tab \[page 470\]](#)

8.5.4.3.4 Group prioritization options: Priority Order tab

These options appear in both the [Match Editor](#) and the [Association Editor](#) based on which transform you are working with.

In the [Post Association Processing](#) or [Post Matching Processing](#) dialog of the `<transform_name> Editor`, select to add a Group Prioritization operation to control group order in post processing.

Group forming prioritization

Use the Group prioritization operation to order records within each break group. This order controls which records are used as the drivers during the comparison process.

Post-match prioritization

Add a Group prioritization operation before a Group Statistics operation to order records within a match group to control which record is flagged as the master record of each group of matching records. Add a Group prioritization operation before a Best Record operation to order records within a match group to control the destination of data that is being propagated from other records to form a best record.

Group Prioritization Editor option description

Option	Description
Prioritization name	Specifies the name for this Group prioritization operation. If you have multiple operations in this Match or Associate transform, be sure to make this name unique.

Priority fields

Use the Priority fields table to order your break groups based on the content of a field (for example, dollar amount or date). Use the buttons to add, remove, and order rows. Place the primary sort field at the top of the list. The rest of the fields, in the order that they are positioned, determine the sub-sort that occurs.

Option	Description
Input field	Choose a field to sort your records on.
Field order	Specifies in which order records should be sorted.

Parent topic: [Post-association processing \[page 457\]](#)

Related Information

Best record options: Best Record tab [page 459]

Best record options: Destination Protection tab [page 462]

Group Statistics Editor options [page 463]

Group prioritization options: Record Completeness tab [page 466]

Unique ID options: Unique ID tab [page 467]

Unique ID options: Destination Protection tab [page 470]

8.5.4.3.5 Group prioritization options: Record Completeness tab

Make settings to set the priority of output data based on record completeness.

Option	Description
<i>Prioritization name</i>	Specifies the name for this Group prioritization operation. If you have multiple operations in this Match transform, be sure to make this name unique.
<i>Order records based on completeness of data</i>	Select this option to apply priority and blank penalty points to records to help control the order of your records.
<i>Define only field penalties</i>	Select this option so the software assesses penalties based on blank fields.
<i>Define priority and penalty fields</i>	Select this option when you have specific fields that contain the actual integer values for priority and blank penalty.
<i>Record priority field</i>	Choose the field that contains priority values. This field must contain an integer.
<i>Apply blank penalty field</i>	Choose the field that contains the indicator (Y or N) for applying blank penalty points to a record.
<i>Define priority and penalty based on input source</i>	Select to have your record priority and blank penalty indicator (Y or N) determined by membership in a given source.
<i>Source Name</i>	Choose an input source from the dropdown list in the Source Name column. The sources listed here are defined in the Input Source operation.
<i>Priority</i>	Type a priority value (an integer) in the Priority column. Remember that the lower the priority score, the higher the priority.
<i>Apply Blank Penalty</i>	Choose Yes or No to determine whether a blank penalty is applied to a record based on membership to this source.
<i>Default record priority</i>	<p>Specifies the default value for the record priority if:</p> <ul style="list-style-type: none">• The record does not contain a field with this value.• The field is blank for a record.• A record does not belong to any of the sources specified. <p>Remember that the lower the priority score, the higher the priority.</p>

Option	Description
<i>Default apply blank penalty</i>	<p>Specifies the default indicator to add blank penalty points to records with blank fields. Use the indicator when a record does not have a field that carries this indicator, if that field is blank or has invalid data, or if a record does not belong to any of the sources specified.</p> <p>Yes: Each blank penalty for a record is added to the record priority to generate an adjusted record priority score. The lower the score, the higher the priority.</p> <p>No: The software does not apply a penalty when the fields are blank.</p>
<i>Input field</i>	Displays the input fields that are available to assign a blank penalty score to.
<i>Blank penalty</i>	Assign a penalty value (an integer) to apply when the specified field is blank in a record.

Parent topic: [Post-association processing \[page 457\]](#)

Related Information

[Best record options: Best Record tab \[page 459\]](#)

[Best record options: Destination Protection tab \[page 462\]](#)

[Group Statistics Editor options \[page 463\]](#)

[Group prioritization options: Priority Order tab \[page 465\]](#)

[Unique ID options: Unique ID tab \[page 467\]](#)

[Unique ID options: Destination Protection tab \[page 470\]](#)

8.5.4.3.6 Unique ID options: Unique ID tab

Use the Unique ID options to assign sequential identification numbers to each new record when adding records to a data warehouse.

For example, the largest number assigned in a particular project can be carried over as the beginning identification number plus 1. The software uses that beginning number in the assignment of new sequential ID numbers. The software assigns new sequential numbers when the software processes the next source against the data warehouse file.

i Note

Also see the Unique ID section for information about working with unique ID in a multiserver environment. Depending on the processing operation and starting value source you use, there could be limitations for using unique ID.

The Unique ID option group includes the following options:

Option	Description
<i>Unique ID name</i>	Enter a name for this Unique ID operation. If you are using other Unique ID operations in this Match transform, distinguish it from other operations. We suggest that you specify the name of the match transform and match level in the unique ID name.
<i>Processing operation</i>	<p>Specifies the type of processing operation you want the application to perform. Valid values include:</p> <p><i>Assign</i>: Assigns a new ID to unique records that need one, or assigns a new ID to all members of a group that don't have an ID. In addition, the assign operation copies an existing ID when a member of a match group already has an ID. For assign operations to work, all match group members must appear consecutively in one collection, and must be in priority order (high to low).</p> <p><i>AssignCombine</i>: Performs both an assign operation and a combine operation. All match group members must appear consecutively in one collection and must be in priority order (high to low).</p> <p><i>Combine</i>: Combines the IDs of a match group when more than one ID is represented. All match group members must appear consecutively in one collection and must be in priority order (high to low).</p> <p><i>Delete</i>: Removes unique IDs from records that have one, unless they are protected.</p> <p><i>Split</i>: Splits the IDs of an ID group when more than one match group is represented. All ID group members must appear consecutively in one collection and must be in priority order (high to low).</p>
<i>Recycle unique IDs</i>	<p>Specifies whether the software should reuse the unique IDs that were freed up during the delete operation in different records. If you have a limited amount of unique ID numbers available, you may want to recycle them. Valid values include:</p> <p><i>Yes</i>: Recycle freed-up unique IDs.</p> <p><i>No</i>: Do not recycle freed-up unique IDs.</p>
<i>ID field</i>	A field that holds a previously assigned unique ID. If this field is omitted, then it is assumed that no records have a unique ID.
<i>Field</i>	<p>The software obtains the starting unique ID from an input field.</p> <p>Be sure to map in a field from an upstream transform before you add this option.</p>
<i>Starting unique ID field</i>	Choose the field that passes in the starting unique ID. If no Unique ID is received, the starting number defaults to 1.
<i>Constant value</i>	The starting ID is specified as a positive whole number in the Starting value option.
<i>Starting value</i>	Indicates the starting unique ID value. Valid values range from 1 to UINT_MAX (unsigned integer max). The default value is 1.
<i>Value from file</i>	The starting Unique ID is read from the file specified in the File option.
<i>File</i>	Specifies the path and name of the file that manages unique IDs. A value is required here only when the Starting unique ID source option is set to File.

Option	Description
<i>GUID</i>	<p>Uses the Globally Unique Identifier (GUID) as the unique ID. GUID is also known as the Universal Unique Identifier (UUID). The UUID variation used for unique ID is a time-based 36-character string with the format: TimeLow-TimeMid-TimeHighAndVersion-ClockSeqAndReservedClockSeqLow-Node.</p> <p>For more information about UUID, see the RFC document in the Related Topics section.</p>
<i>Save ending ID to file and reclaim recycled IDs</i>	<p>Specifies whether to save the last ID that was assigned to a file.</p> <p>Additionally, specifies whether to reclaim recycled IDs.</p>
<i>File</i>	Specifies the file to write the last assigned ID to.
<i>Allow multiple Match transforms to access unique ID file</i>	Allows multiple Match transforms to access a shared unique ID file. When enabled, multiple data flows can access the same unique ID file, and single Match transforms can run in more than one process when the DOP setting is greater than 1. In addition, this option allows multiple Match transforms within a single data flow to share a single unique ID file.
<i>Number of IDs to get when accessing file</i>	<p>Specifies the number of IDs to retrieve from the unique ID file during each access.</p> <p>For example, with a setting of 100, the first process accesses the file and retrieve IDs numbered 1-100. The next process with retrieve IDs numbered 101-200. If a process uses less than the number of retrieved IDs, the remaining IDs are written back to the unique ID file as recycled IDs.</p> <div> <p>i Note</p> <p>A setting greater than 1 improves performance when sharing a unique ID file between multiple processes by reducing the number of times the file must be accessed. However, integer numbers may not be assigned in sequential order.</p> </div>
<i>Group number field</i>	Specifies the field that holds a group number. The group number is used to assign the same unique ID to more than one record. If this field is omitted, then it is assumed that each record is unique and should have its own number.

Parent topic: [Post-association processing \[page 457\]](#)

Related Information

Best record options: [Best Record tab \[page 459\]](#)

Best record options: [Destination Protection tab \[page 462\]](#)

Group Statistics Editor options [\[page 463\]](#)

Group prioritization options: [Priority Order tab \[page 465\]](#)

Group prioritization options: [Record Completeness tab \[page 466\]](#)

Unique ID options: [Destination Protection tab \[page 470\]](#)

8.5.4.3.7 Unique ID options: Destination Protection tab

Use the Destination Protection tab to control whether a record unique ID is protected based on the source that the record belongs to.

This protection can help prevent IDs from being assigned to a suppression or rented source.

Option	Description
<i>Unique ID name</i>	Enter a name for this Unique ID operation. If you are using other Unique ID operations in this Match transform, you may want to specify the name of the match transform and match level in this name to distinguish it from others.
<i>Enable destination protection</i>	Select if you want to protect a destination source from having its unique IDs overwritten with the IDs from matching records.
<i>Default destination protection</i>	Select the default destination protection setting. The protection setting is useful because the default setting accounts for records that are protected (or not protected) through the use of sources or fields.
<i>Specify destination protection by field</i>	Select to enable destination protection through a value in a field.
<i>Unique ID protected field</i>	Choose an input field from the dropdown list that holds the value for specifying whether this ID is protected. The field must contain a value of Y or N. Any other value (including blank) causes the default destination protection specification to occur, if you specified one.
<i>Specify destination protection by source</i>	Select this option to control destination protection through membership in a particular source. Fill in the table with source names and whether they are protected.
<i>Source name</i>	Choose the name of the source from the dropdown list. The list here is populated with defined sources and source groups from the Input Sources Editor window of the Match Editor.
<i>Unique ID protected</i>	Yes: The source is protected. No: The source is not protected.

Parent topic: [Post-association processing \[page 457\]](#)

Related Information

[Best record options: Best Record tab \[page 459\]](#)

[Best record options: Destination Protection tab \[page 462\]](#)

[Group Statistics Editor options \[page 463\]](#)

[Group prioritization options: Priority Order tab \[page 465\]](#)

[Group prioritization options: Record Completeness tab \[page 466\]](#)

[Unique ID options: Unique ID tab \[page 467\]](#)

8.5.4.4 Output fields for the Associate transform

The Associate transform requires that you map one field on output: Group_Number.

Output field name	Default content type	Description
Group_Number	Group_Number	The group number resulting from the association process. Records that belong to the same match group share the same group number. The group numbers start with the number one. Unique records have a blank group number.

Group prioritization output fields

The following output fields are available when you add a Group Prioritization operation to an Associate transform

Output field name	Description
Priority_Value	Record priority value assigned to record. If you did not include a priority value, this field outputs 0.

Parent topic: [Associate \[page 454\]](#)

Related Information

[Associate transform options \[page 455\]](#)


[Association options \[page 456\]](#)

[Post-association processing \[page 457\]](#)

8.5.5 Country ID

The Country ID transform parses your input data and then identifies the country of destination for each record.

Country ID information

Characteristic	Description
	Country ID icon
Use	<p>The Country ID transform parses your input data and then identifies the country of destination for each record. After identifying the country, the transform outputs the country name, any of three different ISO country codes, an ISO script code, and a percentage of confidence in the assignment.</p> <p>Though you can use the Country ID transform before any transform in a data flow, it is most useful during a transactional address cleanse job. Place the Country ID transform before the Global Suggestion List transform. The Global Suggestion List transform needs the ISO_Country_Code_2Char field that the Country ID transform outputs.</p> <p>It is not necessary to use the Country ID transform before the Global Address Cleanse transform in a data flow because the Global Address Cleanse transform contains its own Country ID processing. It is also not necessary to use the Country ID transform before the USA Regulatory Address Cleanse transform because your input data should contain U.S. addresses only.</p>
Content objects	Data Quality provides you with a sample transform that can help you get started creating a Country ID transform useful to you.

[Input fields for the Country ID transform \[page 473\]](#)

[Output fields for the Country ID transform \[page 473\]](#)

Map output fields for the Country ID transform.

Parent topic: [Data Quality transforms \[page 437\]](#)

Related Information

[Blueprints and other content objects for download \[page 438\]](#)

[About Data Quality fields \[page 445\]](#)

[About data quality statistics \[page 448\]](#)

[Associate \[page 454\]](#)
[Data Cleanse \[page 474\]](#)
[DSF2® Walk Sequencer \[page 514\]](#)
[Geocoder \[page 524\]](#)
[Global Address Cleanse \[page 552\]](#)
[Global Suggestion List \[page 640\]](#)
[Match \[page 651\]](#)
[USA Regulatory Address Cleanse \[page 710\]](#)
[Address Cleanse reference \[page 765\]](#)
[Data Cleanse reference \[page 823\]](#)
[Transform configurations \[page 440\]](#)

8.5.5.1 Input fields for the Country ID transform

Use the Input_Fields option group to map the input field that you want to use in this transform.
Here is a list of the Country ID input fields and their descriptions.

Input field name (Country ID)	Description
Country	The country's name or code.
Lastline	The locality, region, and postal code on one line.
Locality1-3	City, town, or suburb information.
Multiline1-12	Lines that may contain any data. The type of data in these lines may vary from record to record.
Postcode	The postal code for the address.
Region1	The state, province, territory, or region of the address.

Parent topic: [Country ID \[page 472\]](#)

Related Information

[Output fields for the Country ID transform \[page 473\]](#)

8.5.5.2 Output fields for the Country ID transform

Map output fields for the Country ID transform.

The following are Data Service output fields that can be defined in the Output tab of the Country ID transform.

Output field name (Country ID)	Description
Confidence_Score	The percentage of certainty that the identified country is accurate. For example, a value of 100 is 100% certainty.
Country_ID_Info_Code	The Country ID info code when the Country ID transform cannot determine a country. <i>1010</i> : Indicates a tie in identifying the country. <i>1005</i> : Indicates that no country was identified.
Country_Name	The identified country name.
ISO_Country_Code_2Char	The 2-character ISO code for the identified country.
ISO_Country_Code_3Char	The 3-character ISO code for the identified country.
ISO_Country_Code_3Digit	The 3-digit ISO code for the identified country.
ISO_Script_Code	The 4-character script code to use for the identified country, such as LATN or KANA.

Parent topic: [Country ID \[page 472\]](#)

Related Information


[Input fields for the Country ID transform \[page 473\]](#)

[Input fields for the Country ID transform \[page 473\]](#)

8.5.6 Data Cleanse

Use the Data Cleanse transform to parse and format custom or person and firm data as well as phone numbers, dates, e-mail addresses, and Social Security numbers.

Data Cleanse information

Characteristic	Description
	Data Cleanse icon

Characteristic	Description
Use	<p>The transform has options to enable generation of data quality statistics tables in the repository. These statistics provide insight into how the software cleansed and assigned your data by generating aggregated summary statistics and specific record-level statistics.</p> <p>The cleansing package that you specify in the transform defines how your data should be parsed and standardized.</p> <div> <p>Note</p> <p>The Social Security number algorithm is no longer updated by the Social Security Administration. The SSN validation performed in Data Cleanse is based on data through July, 2011. SAP builds the Social Security data into the cleansing package.</p> </div> <p>Within a data flow, place the Data Cleanse transform after the address cleansing process and before the matching process.</p>
Content objects	Transform configurations, blueprints, and other content objects.

[Data Cleanse options \[page 476\]](#)

The Data Cleanse transform includes options that control how person, firm, and custom data are parsed and standardized.

[Input fields for the Data Cleanse transform \[page 503\]](#)

Use the Data Cleanse input fields to map fields to use in the transform.

[Output fields for the Data Cleanse transform \[page 505\]](#)

Use the output fields to map output from the Data Cleanse transform.

Parent topic: [Data Quality transforms \[page 437\]](#)

Related Information

[Blueprints and other content objects for download \[page 438\]](#)

[About Data Quality fields \[page 445\]](#)

[About data quality statistics \[page 448\]](#)

[Associate \[page 454\]](#)

[Country ID \[page 472\]](#)

[DSF2® Walk Sequencer \[page 514\]](#)

[Geocoder \[page 524\]](#)

[Global Address Cleanse \[page 552\]](#)

[Global Suggestion List \[page 640\]](#)
[Match \[page 651\]](#)
[USA Regulatory Address Cleanse \[page 710\]](#)
[Address Cleanse reference \[page 765\]](#)
[Data Cleanse reference \[page 823\]](#)
[Data Quality Statistics Settings \[page 479\]](#)

8.5.6.1 Data Cleanse options

The Data Cleanse transform includes options that control how person, firm, and custom data are parsed and standardized.

[Common options \[page 477\]](#)

There is one common option in the Data Cleanse transform: *Run as Separate Process*.

[Report and Analysis \[page 478\]](#)

Select options for generating data quality statistics and select to generate report data for the Data Cleanse transform.

[Data Quality Statistics Settings \[page 479\]](#)

The Data Quality Statistics Settings control the generation of data quality statistics tables.

[Cleansing Package options \[page 482\]](#)

Name a cleansing package to use and set other options related to the cleansing package.

[Options \[page 485\]](#)

The Options group includes settings that control how the Data Cleanse transform parses and outputs data.

[Input word breaker \[page 486\]](#)

Options in the input word breaker group control how the parser breaks input data.

[Person standardization options \[page 487\]](#)

The options in the Person group control how the transform standardizes person-related output.

[Gender standardization options \[page 491\]](#)

The gender standardization options control which input fields Data Cleanse uses to assign gender.

[Firm standardization options \[page 492\]](#)

The firm standardization options control how the Data Cleanse transform standardizes firm-related output.

[Other standardization options \[page 493\]](#)

Standardization options control how the Data Cleanse transform standardizes many types of output.

[Date options \[page 496\]](#)

Configure standards for date data such as the date format, date delimiter, and so on.

[Phone Options group \[page 499\]](#)

Set options for international and North American phone parsing.

[Parser configuration \[page 501\]](#)

Parser configuration options control which parsing engines Data Cleanse uses for parsing multiline fields and the order in which they are applied.

[Memory cache \[page 502\]](#)

Memory cache allocates memory per thread for processing the Data Cleanse transform.

Parent topic: [Data Cleanse \[page 474\]](#)

Related Information

[Input fields for the Data Cleanse transform \[page 503\]](#)

[Output fields for the Data Cleanse transform \[page 505\]](#)

8.5.6.1.1 Common options

There is one common option in the Data Cleanse transform: *Run as Separate Process*.

Option	Description
Run As Separate Process	<i>Yes</i> : Splits the transform into a separate process. <i>No</i> : Keeps the transform in the same process as the rest of the data flow.

Parent topic: [Data Cleanse options \[page 476\]](#)

Parent topic: [DSF2® Walk Sequencer \[page 514\]](#)

Related Information

[Report and Analysis \[page 478\]](#)

[Data Quality Statistics Settings \[page 479\]](#)

[Cleansing Package options \[page 482\]](#)

[Options \[page 485\]](#)

[Input word breaker \[page 486\]](#)

[Person standardization options \[page 487\]](#)

[Gender standardization options \[page 491\]](#)

[Firm standardization options \[page 492\]](#)

[Other standardization options \[page 493\]](#)

[Date options \[page 496\]](#)

[Phone Options group \[page 499\]](#)

[Parser configuration \[page 501\]](#)

[Memory cache \[page 502\]](#)

[Reference Files \[page 517\]](#)

[Processing Options \[page 517\]](#)
[USPS License Information \[page 518\]](#)
[Data Collection Config \[page 519\]](#)
[Input fields for the DSF2 Walk Sequencer transform \[page 520\]](#)
[Output fields for the DSF2 Walk Sequencer transform \[page 522\]](#)

8.5.6.1.2 Report and Analysis

Select options for generating data quality statistics and select to generate report data for the Data Cleanse transform.

Option	Description
Data Quality Statistics Settings	Settings that control the generation of data quality statistics tables.
Generate Report Data	<p>Specifies whether to generate report data for this transform.</p> <p>Yes: Generates report data for this transform.</p> <p>No: Turns off report data generation. If you do not need to generate reports (during testing, for example), you should set this option to No to improve performance.</p>

Parent topic: [Data Cleanse options \[page 476\]](#)

Related Information

[Common options \[page 477\]](#)
[Data Quality Statistics Settings \[page 479\]](#)
[Cleansing Package options \[page 482\]](#)
[Options \[page 485\]](#)
[Input word breaker \[page 486\]](#)
[Person standardization options \[page 487\]](#)
[Gender standardization options \[page 491\]](#)
[Firm standardization options \[page 492\]](#)
[Other standardization options \[page 493\]](#)
[Date options \[page 496\]](#)
[Phone Options group \[page 499\]](#)
[Parser configuration \[page 501\]](#)
[Memory cache \[page 502\]](#)

8.5.6.1.3 Data Quality Statistics Settings

The Data Quality Statistics Settings control the generation of data quality statistics tables.

The [Data Quality Statistics Settings](#) options in the following table apply to either Data Cleanse, Global Address Cleanse, or both transforms. Some also apply to the Geocoder transform. The following table includes a column that indicates the applicable transforms.

Data quality statistics option descriptions

Option	Transform	Description
Generate Cleanse Statistics Table	Data Cleanse	Specifies to generate the Cleanse Statistics table. Options include:
	Global Address Cleanse	<ul style="list-style-type: none">Yes: The software populates the CLEANSE_STATISTICS_ table with a row of statistics for each unique ENTITY_ID.No: The software does not populate the CLEANSE_STATISTICS_ table. This is the default setting. <p>The Cleanse Statistics table contains summarized data for each record with a unique value in the <code><ENTITY_ID></code> field. Use the data to gain a high-level insight into what the software changed during the cleansing process.</p> <p>Information in the table includes the total number of:</p> <ul style="list-style-type: none">RecordsSuspect recordsBlank recordsRecords with one or more significant change
Generate Cleanse Info Codes Table	Data Cleanse	Specifies to populate the Cleanse Info Codes table in the repository. Options include:
	Global Address Cleanse	<ul style="list-style-type: none">Yes: The software populates the CLEANSE_INFO_CODES_ table in the repository with a row for each significant information code generated by the cleanse process.No: The software does not populate the CLEANSE_INFO_CODES_ table. This is the default setting. <p>Analyze this table to find potential suspect data in your data source. Analyze the table information to view a distribution of information codes or a count of missing, suspect, or blank data.</p> <div>i Note Not all existing information codes apply to data quality statistics.</div>

Option	Transform	Description
Generate Cleanse Change Info Table	Data Cleanse	<p>Specifies to populate the Cleanse Change Info table in the repository. Options include:</p> <ul style="list-style-type: none"> • Yes: The software populates the CLEANSE_CHANGE_INFO_ table in the repository with a row for each significant change applied by the cleansing operation. • No: The software does not populate the CLEANSE_CHANGE_INFO_ table. This is the default setting. <p>Analyze this detailed table to identify concerning areas of your source data by the number of significant changes.</p> <div> ❖ Example You research a data source further when the data shows a high count of significant changes for firm or address components. </div>
	Global Address Cleanse	
Generate Cleanse Component Info Table	Data Cleanse	<p>Specifies to populate the Cleanse Component Info table. Options include:</p> <ul style="list-style-type: none"> • Yes: The software populates the CLEANSE_COMPONENT_INFO_ table in the repository. • No: The software does not populate the CLEANSE_COMPONENT_INFO_ table. This is the default setting. <p>This detailed table contains position information for each specified data element under the following circumstances:</p> <ul style="list-style-type: none"> • When it is parsed from an input field. • When it is written to an output field. • When it is parsed from an input field and written to an output field. <p>The transform can populate an output field column when there is no direct link to an input. And an input may not have any directly linked output.</p> <div> ❖ Example Two examples: <ul style="list-style-type: none"> • On input there is no prename included for a name. However, on output, the prename is included because of reference data. • On input, the transform maps to <code><given_name1></code>. And, on output, the transform does not include any <code><person1></code> output fields. </div>
	Global Address Cleanse	

Option	Transform	Description
Generate Cleanse Address Record Info Table	Global Address Cleanse	<p>Applies to Global Address Cleanse for address data. Specifies to populate the Cleanse Address Record Info table in the repository. Options include:</p> <ul style="list-style-type: none"> Yes: The software populates the CLEANSE_ADDRESS_RECORD_INFO_ table in the repository with up to two rows per record for each cleansed address entity. No: The software does not populate the CLEANSE_ADDRESS_RECORD_INFO_ table. This is the default setting. <p>Analyze this detailed table to gain information about the results of the address cleanse process on the data. The table contains statistics that focus on the results of address cleansing, which includes assignment information and assignment type for each address.</p>
Generate Geocode Statistics Table	Geocoder	<p>Specifies to populate the Geocode Statistics table in the repository. Options include:</p> <ul style="list-style-type: none"> Yes: The software populates the GEOCODE_STATISTICS_ table in the repository with one row of totals for each job execution. No: The software does not populate the GEOCODE_STATISTICS_ table. This is the default setting. <p>Analyze this table for a summary of the geocode process on the data. The table provides the total number of rows in the data and the total number of rows that were assigned a latitude and longitude.</p>
Generate Geocode Info Codes Table	Geocoder	<p>Specifies to populate the Geocode Info Codes table in the repository. Options include:</p> <ul style="list-style-type: none"> Yes: The software populates the GEOCODE_INFO_CODES_ table in the repository with one row for each significant information code assigned during the geocode process. No: The software does not populate the GEOCODE_INFO_CODES_ table. This is the default setting. <p>Use the information codes to analyze this detailed table. Track the reasons why records were not assigned the highest level of latitude and longitude.</p> <div> <p>i Note</p> <p>Not all existing information codes apply to the data quality statistics.</p> </div>

Parent topic: [Data Cleanse options \[page 476\]](#)

Related Information

[Common options \[page 477\]](#)
[Report and Analysis \[page 478\]](#)
[Cleansing Package options \[page 482\]](#)
[Options \[page 485\]](#)
[Input word breaker \[page 486\]](#)
[Person standardization options \[page 487\]](#)
[Gender standardization options \[page 491\]](#)
[Firm standardization options \[page 492\]](#)
[Other standardization options \[page 493\]](#)
[Date options \[page 496\]](#)
[Phone Options group \[page 499\]](#)
[Parser configuration \[page 501\]](#)
[Memory cache \[page 502\]](#)
[Data quality statistics common requirements \[page 449\]](#)
[Data quality statistics transform set up requirements \[page 451\]](#)
[Content information for data quality statistics tables \[page 1346\]](#)

8.5.6.1.4 Cleansing Package options

Name a cleansing package to use and set other options related to the cleansing package.

Data Cleanse transform options.

Data Cleanse cleansing package option descriptions

Option	Description
<i>Cleansing Package Name</i>	Name of the cleansing package that the Data Cleanse transform uses to parse data.

Option	Description
<i>Content Domain Sequence</i>	<p>A content domain specifies the specific domain properties to assign to a variation. You can specify more than one content domain.</p> <p>The Global domain is a special content domain, which contains all variations and their associated properties. If a variation is not associated with domain-specific information the Global domain serves as the default domain. The Global domain is required for every content domain sequence. Be sure to add GLOBAL as the last domain in the sequence.</p> <div> <p>Note</p> <p>You can set this option as a dynamic input field.</p> </div> <p>Select the content domains you want to include. The arrows allow you to change the order of the content domains.</p> <p>GLOBAL - Global AR - Arabic ZH - Chinese CS - Czech DA - Danish NL - Dutch EN_US - English (United States & Canada) EN_GB - English (United Kingdom & Ireland) EN_AU - English (Australia & New Zealand) EN_IN - English (India) FR - French DE - German HU - Hungarian ID - Indonesian IT - Italian JA - Japanese MS - Malay NO - Norwegian PL - Polish PT_BR - Portuguese (Brazil) PT_PT - Portuguese (Portugal) RO - Romanian RU - Russian SK - Slovak ES_MX - Spanish (Latin America) ES_ES - Spanish (Spain) SV - Swedish TR - Turkish</p>

Option	Description
<i>Output Format</i>	<p>Selects the format for output. Based on the specified domain in the output format. Data Cleanse uses certain output fields and formats from the data in those fields according to the regional standards.</p> <div> <p>Note</p> <p>You can set this option as a dynamic input field.</p> </div> <p>Valid values for this option are:</p> <ul style="list-style-type: none"> AR Arabic ZH Chinese CS Czech DA Danish NL Dutch EN_US English (United States & Canada) EN_GB English (United Kingdom & Ireland) EN_AU English (Australia & New Zealand) EN_IN English (India) FR French DE German HU Hungarian ID Indonesian IT Italian JA Japanese MS Malay NO Norwegian PL Polish PT_BR Portuguese (Brazil) PT_PT Portuguese (Portugal) RO Romanian RU Russian SK Slovak ES_MX Spanish (Latin America) ES_ES Spanish (Spain) SV Swedish TR Turkish

Parent topic: [Data Cleanse options \[page 476\]](#)

Related Information

[Common options \[page 477\]](#)

[Report and Analysis \[page 478\]](#)

[Data Quality Statistics Settings \[page 479\]](#)

[Options \[page 485\]](#)

[Input word breaker \[page 486\]](#)
[Person standardization options \[page 487\]](#)
[Gender standardization options \[page 491\]](#)
[Firm standardization options \[page 492\]](#)
[Other standardization options \[page 493\]](#)
[Date options \[page 496\]](#)
[Phone Options group \[page 499\]](#)
[Parser configuration \[page 501\]](#)
[Memory cache \[page 502\]](#)

8.5.6.1.5 Options

The Options group includes settings that control how the Data Cleanse transform parses and outputs data.

The following list contains the groups of options in the Options group:

- [Input Word Breaker](#)
- [Standardization Options](#)
- [Date Options](#)
- [Phone Options](#)
- [Parser Configuration](#)

Find descriptions for the options in these groups in the applicable topic. In the application, click the group to see the descriptions in the help pane at right.

Option	Description
Filter Output Fields	<p>Specifies which output fields are displayed in the Output tab.</p> <p>Show_Relevant_Fields: The fields available in the Output tab are based on the mapped input fields and the selected parser sequence multiline options. Includes all output fields that could possibly contain parsed data. The <code><Extra></code> fields are always available.</p> <p>Show_All_Fields: All Data Cleanse transform fields are available.</p>
Memory in KB for Cache	<p>(Optional) Enter a value of memory to allocate per thread for processing the Data Cleanse transform. Setting this option to 0 indicates that the internal default value (65536) should be used.</p>

Parent topic: [Data Cleanse options \[page 476\]](#)

Related Information

[Common options \[page 477\]](#)
[Report and Analysis \[page 478\]](#)

[Data Quality Statistics Settings \[page 479\]](#)
[Cleansing Package options \[page 482\]](#)
[Input word breaker \[page 486\]](#)
[Person standardization options \[page 487\]](#)
[Gender standardization options \[page 491\]](#)
[Firm standardization options \[page 492\]](#)
[Other standardization options \[page 493\]](#)
[Date options \[page 496\]](#)
[Phone Options group \[page 499\]](#)
[Parser configuration \[page 501\]](#)
[Memory cache \[page 502\]](#)

8.5.6.1.6 Input word breaker

Options in the input word breaker group control how the parser breaks input data.

Option	Description
Break on Whitespace Only	<p>Specifies whether the Data Cleanse transform breaks input data only on white space or on white space, punctuation, alphanumeric transitions, and script transitions.</p> <p>Yes: Input data breaks only on white space.</p> <p>No: Input data breaks on white space, punctuation, alphanumeric transitions, and script transitions. For data in the CJK and Kana scripts, input data breaks on each character.</p> <p>This option allows the Data Cleanse transform to recognize alphanumeric product codes as entries in a custom cleansing package. For example, if <i>Break on Whitespace only</i> is set to no, the parser breaks a product code such as AF302 into two tokens, AF and 302. If <i>Break on Whitespace only</i> is set to yes, the parser recognizes AF302 as a single entry.</p> <div><p>Note</p><p>This option typically applies to custom cleansing packages. The out of the box person and firm cleansing packages are designed to use the parsing strategy that breaks data on white space, punctuation, alphanumeric transitions, and script transitions.</p></div>

Parent topic: [Data Cleanse options \[page 476\]](#)

Related Information

[Common options \[page 477\]](#)
[Report and Analysis \[page 478\]](#)
[Data Quality Statistics Settings \[page 479\]](#)
[Cleansing Package options \[page 482\]](#)
[Options \[page 485\]](#)
[Person standardization options \[page 487\]](#)
[Gender standardization options \[page 491\]](#)
[Firm standardization options \[page 492\]](#)
[Other standardization options \[page 493\]](#)
[Date options \[page 496\]](#)
[Phone Options group \[page 499\]](#)
[Parser configuration \[page 501\]](#)
[Memory cache \[page 502\]](#)

8.5.6.1.7 Person standardization options

The options in the Person group control how the transform standardizes person-related output.

Note

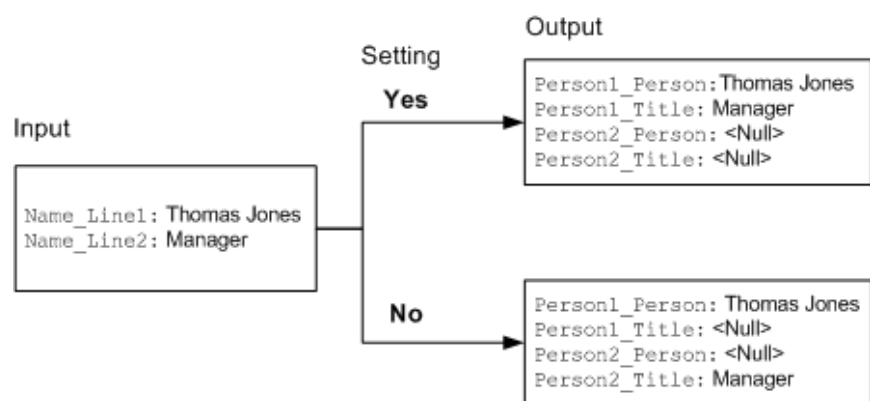
The options in this group apply only to person and firm cleansing packages.

Find descriptions for the [Gender Options](#) group in the applicable topic. If you are in the application, click the group name and the option descriptions appear in the Help pane at right.

Options under Standardization Options

Option	Description
Assign Prenames	<p>Specifies whether the transform should include assigned prenames (for example, Mr. or Mrs.) in the <code><Prenames></code> output field.</p> <p>Yes: Turns on prename assignment.</p> <p>No: Turns off prename assignment. The <code><Prenames></code> output field contains only prenames included in the input data</p> <p>The <code><Prenames></code> output field always includes prenames that are part of the name input data. Additionally, the Data Cleanse transform can assign prenames based on the gender of the name (strong_male or strong_female) in the <code><Given_Name1></code> field. When the gender of Given_Name1 is not strong, prenames are assigned based on the settings for the Gender Options > Use Given Name2 To Assign Gender and Gender Options > Use Family Name to Assign Gender options.</p>

Option	Description
Associate Name Title	<p>Defines how name and occupational title data found in separate input fields are associated.</p> <p>Yes: Data Cleanse assumes that the name and title data describe the same person and is associated.</p> <p>No: Data Cleanse assumes that the name and title data is not associated.</p> <p>For example, the following diagram shows the difference in the output based on the <i>Associate Name Title</i> setting.</p>



i Note

If you are generating data quality statistics tables in the Data Cleanse transform, set this option to *No*.

Combine Compound Names	<p>Specifies how compound family names are standardized when the family name includes a Pre_Family_Name that is also a Pre_Family_Name_Combine.</p> <p>Yes: Combines compound family names. For example, the family name Mc Donald would combine to McDonald.</p> <p>No: Retains a space between the Pre_Family_Name and the family name. For example, the family name Mc Donald remains Mc Donald.</p>
------------------------	---

i Note

This option has no impact on a Pre_Family_name that is not classified as a Pre_Family_Name_Combine. For example, the software does not combine the name Van Helsing into VanHelsing because "Van" is not classified with both PRE_FAMILY_NAME and PRE_FAMILY_NAME_COMBINE.

Option	Description
Enable Presumptive Name Parsing	<p>Specifies whether you want to use presumptive name parsing on Name_Line input fields.</p> <p>Yes: Turns on presumptive name parsing. Data in the <code><Name_Line></code> input field is treated as a name.</p> <p>No: Turns off presumptive name parsing. Data in the <code><Name_Line></code> input field that does not parse as a name remains unparsed and is output to the <code><Extra></code> field.</p> <div> <p>❖ Example</p> <p>If the data contains an automobile brand and model in a <code><Name_Line></code> input field, the Data Cleanse transform tries to parse the information as a name based on rules in the Cleansing Package. If the option is set to No and Data Cleanse is not able to assign the data, the unparsed data is output to the <code><Extra></code> field. If the option is set to Yes, Data Cleanse assigns the data as a name.</p> </div>
Name Order	<p>Defines how Data Cleanse applies parsing rules to determine the content of the Given_Name and Family_Name output fields.</p> <p>Given_Family_Name_Strict and Family_Given_Name_Strict: These values specify the respective order of given and family names in the input file. Parsing rules that do not follow the strictly defined name order are not considered when Data Cleanse determines which rule to apply to the input string.</p> <p>These settings are useful when the order of the family and given names in the input data is consistent.</p> <p>Given_Family_Name_Suggest and Family_Given_Name_Suggest: These values specify which rule to choose to break a tie when two rules have the same confidence score. Data Cleanse chooses the rule that follows the suggested name order.</p> <p>Unknown: Data Cleanse chooses the rule with the highest confidence score based on information in the Cleansing Package. In the case of a tie, Data Cleanse chooses the first rule in the rule order.</p>

Option	Description																								
Parse Discrete Input	<p>Defines how to parse person data.</p> <p>No: discrete input fields are mapped directly to the corresponding output fields without being parsed.</p> <p>Yes: Discrete input fields are combined to one input field so the data can be parsed and output to discrete fields.</p> <div><div><div>❖ Example</div><div>Input data</div><table><tr><th>Column</th><th>Field</th></tr><tr><td>Person1_Given_Name1</td><td>Mr John T</td></tr><tr><td>Person1_Family_Name1</td><td>Smith Iii</td></tr></table><div>Output data</div><table><tr><th>Column</th><th>Option=No</th><th>Option=Yes</th></tr><tr><td>Person1.Prename</td><td><blank></td><td>Mr</td></tr><tr><td>Person1.Given_Name1</td><td>Mr John T</td><td>John</td></tr><tr><td>Person1.Given_Name2</td><td><blank></td><td>T</td></tr><tr><td>Person1.Family_Name1</td><td>Smith Iii</td><td>Smith</td></tr><tr><td>Person1.Maturity_Post-name</td><td><blank</td><td>III</td></tr></table></div></div>	Column	Field	Person1_Given_Name1	Mr John T	Person1_Family_Name1	Smith Iii	Column	Option=No	Option=Yes	Person1.Prename	<blank>	Mr	Person1.Given_Name1	Mr John T	John	Person1.Given_Name2	<blank>	T	Person1.Family_Name1	Smith Iii	Smith	Person1.Maturity_Post-name	<blank	III
Column	Field																								
Person1_Given_Name1	Mr John T																								
Person1_Family_Name1	Smith Iii																								
Column	Option=No	Option=Yes																							
Person1.Prename	<blank>	Mr																							
Person1.Given_Name1	Mr John T	John																							
Person1.Given_Name2	<blank>	T																							
Person1.Family_Name1	Smith Iii	Smith																							
Person1.Maturity_Post-name	<blank	III																							

Parent topic: [Data Cleanse options \[page 476\]](#)

Related Information

[Common options \[page 477\]](#)

[Report and Analysis \[page 478\]](#)

[Data Quality Statistics Settings \[page 479\]](#)

[Cleansing Package options \[page 482\]](#)

[Options \[page 485\]](#)

[Input word breaker \[page 486\]](#)

[Gender standardization options \[page 491\]](#)

[Firm standardization options \[page 492\]](#)

[Other standardization options \[page 493\]](#)

[Date options \[page 496\]](#)

[Phone Options group \[page 499\]](#)

[Parser configuration \[page 501\]](#)

8.5.6.1.8 Gender standardization options

The gender standardization options control which input fields Data Cleanse uses to assign gender.

Find these options in the [Gender Options](#) group under [Options](#) > [Standardization Options](#) > [Person](#) > [Person](#).

Note

The options in this group apply only to person and firm cleansing packages.

Option	Description
Use Given Name2 To Assign Gender (Gender Options)	<p>When the gender of the prename and Given_Name1 are unassigned or ambiguous, assigns gender based on the gender of the parsed Given_Name2.</p> <p>Yes: Turns on the option.</p> <p>No: Turns off the option.</p> <div><p>❖ Example</p><p>if this option is set to No, the gender of the name Pat Robert Smith is ambiguous because the Given_Name1, Pat, is ambiguous. However, if the option is set to Yes, the gender is Strong_Male because the Given_Name2, Robert, is Strong_Male. The same logic applies if the name were P. Robert Smith; the Given_Name1, P, is ambiguous.</p></div>
Use Family Name To Assign Gender (Gender Options)	<p>When the gender of the prename, Given_Name1, and Given_Name2 are unassigned or ambiguous, assigns gender based on the gender of the family name. Uses Family_Name1 if gender is assigned and is not ambiguous. Uses Family_Name2 if unable to use Family_Name1.</p> <p>Yes: Turns on the option.</p> <p>No: Turns off the option.</p> <p>For example, if the option is set to No, the gender of the name N. Albiantsev is ambiguous because the Given_Name1, N., is ambiguous. However, if the option is set to Yes, the gender is Strong_Male because the Family_Name1, Albiantsev is Strong_Male.</p>

Parent topic: [Data Cleanse options \[page 476\]](#)

Related Information

[Common options \[page 477\]](#)

[Report and Analysis \[page 478\]](#)

[Data Quality Statistics Settings \[page 479\]](#)

[Cleansing Package options \[page 482\]](#)
[Options \[page 485\]](#)
[Input word breaker \[page 486\]](#)
[Person standardization options \[page 487\]](#)
[Firm standardization options \[page 492\]](#)
[Other standardization options \[page 493\]](#)
[Date options \[page 496\]](#)
[Phone Options group \[page 499\]](#)
[Parser configuration \[page 501\]](#)
[Memory cache \[page 502\]](#)

8.5.6.1.9 Firm standardization options

The firm standardization options control how the Data Cleanse transform standardizes firm-related output.

i Note

The options in this group apply only to person and firm cleansing packages.

Option	Description
Enable Presumptive Firm Parsing	<p>Specifies whether you want to use presumptive firm parsing on Firm_Line input fields.</p> <p>Yes: Turns on presumptive firm parsing. Data in the Firm_Line input field is treated as a firm name.</p> <p>No: Turns off presumptive firm parsing. Data in the Firm_Line input field that does not parse as a firm remains unparsed and is output to the Extra field.</p> <p>For example, if the data has a given name and family name in a Firm_Line input field, the Data Cleanse transform tries to parse the information as a firm based on rules in the cleansing package. If the option is set to No and Data Cleanse is not able to assign the data, the unparsed data is output to the Extra field. If the option is set to Yes, Data Cleanse will assign the data as a firm.</p>

Parent topic: [Data Cleanse options \[page 476\]](#)

Related Information

[Common options \[page 477\]](#)
[Report and Analysis \[page 478\]](#)
[Data Quality Statistics Settings \[page 479\]](#)
[Cleansing Package options \[page 482\]](#)
[Options \[page 485\]](#)

[Input word breaker \[page 486\]](#)
[Person standardization options \[page 487\]](#)
[Gender standardization options \[page 491\]](#)
[Other standardization options \[page 493\]](#)
[Date options \[page 496\]](#)
[Phone Options group \[page 499\]](#)
[Parser configuration \[page 501\]](#)
[Memory cache \[page 502\]](#)

8.5.6.1.10 Other standardization options

Standardization options control how the Data Cleanse transform standardizes many types of output.

Option	Description
Capitalization	<p>Specifies the casing of your output.</p> <p><i>Lower</i>: Converts the output to lowercase. For example, john mckay.</p> <p><i>Mixed</i>: Preserves the casing for the standard form as defined within the cleansing package. If a standard form is not defined, the output is converted to mixed case.</p> <p>For example, if the standard form is defined as John Mckay, it is preserved. If a standard form is not defined, the output is converted to mixed case, John McKay.</p> <p><i>Preserve</i>: Preserves the input casing.</p> <p><i>Upper</i>: Converts the output to uppercase. For example, JOHN MCKAY.</p>

Option	Description
Character Width Style	<p>Specifies the character width used in output fields. Useful when processing Japanese or mixed language data.</p> <p><i>Normal_Width</i>: Output field width reflects the normalized character width based on the script type. Thus some output columns contain half-width characters and other columns contain full-width characters. For example, all full-width Latin characters are standardized to their half-width forms and all half-width katakana characters are standardized to their full-width forms. <i>Normal_Width</i> does not require special processing and therefore is the most efficient setting.</p> <p><i>Full-width</i>: Characters are converted from their half-width forms to full-width forms for all output fields. For characters that do not have full-width forms, the half-width forms are used.</p> <p><i>Half-width</i>: Characters are converted from their full-width forms to half-width forms for all output fields. For characters that do not have half-width forms, the full-width forms are used.</p> <div> <p>Note</p> <p>Since the output width is based on the normalized width for the character type, the output data may be larger than the input data. You may need to increase the column width in the target table.</p> </div> <p>For template tables, selecting the <i>Use NVARCHAR for VARCHAR columns in supported databases</i> box changes the VARCHAR column type to NVARCHAR and allows for increased data size.</p>
One-to-one mapping	<p>Specifies whether to place the input data into the corresponding output field for the following parsers: Phone, E-mail, Date.</p> <p><i>Yes</i>: Places the parsed data into the corresponding output field. For example, if on input, Date1 and Date2 are blank and Date3 contains data, then on output, Date1 and Date2 are blank and the software places the data in Date3.</p> <p><i>No</i>: Places the parsed data into the first available output field in the category. For example, if on input Date1 and Date2 are blank and Date3 contains data, then on output, Date1 contains the parsed data that was input in the Date3 field.</p>
Remove Diacritical Characters	<p>Removes diacritical characters and replaces it with the ASCII equivalent.</p> <p><i>Yes</i>: Replaces diacritical characters such as accent marks, umlauts, and so on, with the ASCII equivalent.</p> <p><i>No</i>: Retains the standardized diacritical characters.</p> <p>For example, when the option is set to <i>No</i>, the data is output with accent marks such as María Hernández or Geschäftsführer. When the option is set to <i>Yes</i>, the data is output without accent marks such as Maria Hernandez or Geschaeftsfuehrer.</p>

Option	Description
Remove Punctuation	<p>Removes all punctuation from standardized data (with the exception of hyphens between names).</p> <p><i>Yes:</i> Removes punctuation.</p> <p><i>No:</i> Leaves the punctuation as is on input.</p> <p>For example, if the standard form for extra large is X.L. and the option is set to Yes, the standardized output becomes XL.</p>
SSN Delimiter	<p>Specifies which character to use for standard U.S. Social Security number (SSN) output delimiters.</p> <p><i>Backslash (\):</i> Uses backward slashes as the delimiter in the SSN. For example, 799\45\6789.</p> <p><i>Dash (-):</i> Uses dashes as the delimiter in the SSN. For example, 799-45-6789.</p> <p><i>Slash (/):</i> Uses forward slashes as the delimiter in the SSN. For example, 799/45/6789.</p> <p><i>None:</i> Does not add a delimiter to the SSN. For example, 799456789.</p> <p><i>Period (.):</i> Uses periods as the delimiter in the SSN. For example, 799.45.6789.</p> <p><i>Space:</i> Uses spaces as the delimiter in the SSN. For example, 799 45 6789.</p>

Parent topic: [Data Cleanse options \[page 476\]](#)

Related Information

[Common options \[page 477\]](#)
[Report and Analysis \[page 478\]](#)
[Data Quality Statistics Settings \[page 479\]](#)
[Cleansing Package options \[page 482\]](#)
[Options \[page 485\]](#)
[Input word breaker \[page 486\]](#)
[Person standardization options \[page 487\]](#)
[Gender standardization options \[page 491\]](#)
[Firm standardization options \[page 492\]](#)
[Date options \[page 496\]](#)
[Phone Options group \[page 499\]](#)
[Parser configuration \[page 501\]](#)
[Memory cache \[page 502\]](#)

8.5.6.1.11 Date options

Configure standards for date data such as the date format, date delimiter, and so on.

Date option descriptions

Option group	Description
Century Threshold	<p>Indicates whether a two-digit date is considered part of the 20th or 21st century. The default value is 25.</p> <p>Specify a two-digit integer that represents the first year that a parsed two-digit year is considered part of the 21st century (20xx). All two-digit years greater than the specified integer are considered part of the 20th century (19xx).</p> <p>For example, if you enter 11, all two-digit years 11 or lower are considered part of the 21st century. 08 is considered 2008. 11 is considered 2011. All two-digit years higher than 11 are considered part of the 20th century. 12 is considered 1912.</p>
Date Delimiter	<p>Specifies what character to use for standard date output delimiters.</p> <p><i>Backslash (\)</i>: Uses backward slashes as the delimiter for the date. For example, 04\01\2010.</p> <p><i>Dash (-)</i>: Uses dashes as the delimiter for the date. For example, 04-01-2010.</p> <p><i>Slash (/)</i>: Uses forward slashes as the delimiter for the date. For example, 04/01/2010.</p> <p><i>None</i>: Does not add a delimiter to the date. For example, 04012010</p> <p><i>Period (.)</i>: Uses periods as the delimiter for the date. For example, 04.01.2010.</p> <p><i>Space</i>: Uses spaces as the delimiter for the date. For example, 04 01 2010.</p> <p><i>Chinese_Japanese</i>: Uses the following Chinese or Japanese characters as delimiters:</p> <ul style="list-style-type: none">• 月 always follows the month• 日 always follows the day• 年 always follows the year <p>An example of Arabic numbers with Chinese or Japanese delimiters is:</p> <p>04 月 01 日 2010 年</p> <p>An example of Chinese or Japanese Numbers with Chinese or Japanese delimiters is:</p> <p>四 月 一 日 二 千 零 一 十 年</p>
Date Format	<p>Specifies how to standardize date output.</p> <p>YEAR_MONTH_DAY: For Example, 2012-08-16</p> <p>YEAR_DAY_MONTH: For Example, 2012-16-08</p> <p>MONTH_DAY_YEAR: For Example, 08-16-2012</p> <p>DAY_MONTH_YEAR: For Example, 16-08-2012</p>

Option group	Description
Enable Zero Pad	<p>Specifies placement of a zero on the front of one-digit days and months. For example, July 4 could be 04-07 (or 07-04) with a zero pad, and 4-7 (or 7-4) without a zero pad.</p> <p><i>Yes:</i> Turns on the option.</p> <p><i>No:</i> Turns off the option.</p>
Input Month Before Day	<p>Specifies whether the date follows the pattern of having the month first or the day first in the input.</p> <p><i>Yes:</i> The month is first. For example, 11/12/2004 would be November 12, 2004.</p> <p><i>No:</i> The day is first. For example, 11/12/2004 would be December 11, 2004.</p>
Input Year First	<p>Specifies whether the date follows the pattern of having the year first in the input.</p> <p><i>Yes:</i> The year is first. For example, if your input is 03/02/04, the transform converts it to 2003 February 4.</p> <p><i>No:</i> The month is first. For example, if your input is 03/02/04 the transform converts it to March 2, 2004.</p>
Month Format	<p>Specifies how to standardize date and month components.</p> <p><i>Full_Text:</i> Standardizes output with spelled-out months. The language of the month is based on the domain selected in the Output Format option. For example, if you select one of the English domains, the full-text month is January, February, March, and so on.</p> <p><i>Numeric:</i> Standardizes output with numeric months (for example, 03).</p> <p><i>Short_Text:</i> Standardizes output with abbreviated months. The language of the month is based on the domain selected in the Output Format option. For example, if you select one of the English domains, the short-text month is Jan., Feb., Mar., and so on.</p>
Numeric Format	<p>Specifies the format of numeric date values</p> <p><i>Arabic_Numbers:</i> Returns numeric date values in Arabic</p> <p><i>Chinese_Japanese_Numbers:</i> Returns numeric date values in Chinese or Japanese.</p>
Year Format	<p>Specifies how to standardize date and year components.</p> <p><i>Full:</i> Standardizes output with four-digit years (for example, 2004).</p> <p><i>Short:</i> Standardizes output with two-digit years (for example, 04).</p>

Parent topic: [Data Cleanse options \[page 476\]](#)

Related Information

[Common options \[page 477\]](#)

[Report and Analysis \[page 478\]](#)

[Data Quality Statistics Settings \[page 479\]](#)

[Cleansing Package options \[page 482\]](#)

[Options \[page 485\]](#)
[Input word breaker \[page 486\]](#)
[Person standardization options \[page 487\]](#)
[Gender standardization options \[page 491\]](#)
[Firm standardization options \[page 492\]](#)
[Other standardization options \[page 493\]](#)
[Phone Options group \[page 499\]](#)
[Parser configuration \[page 501\]](#)
[Memory cache \[page 502\]](#)

8.5.6.1.12 Phone Options group

Set options for international and North American phone parsing.

Phone Options descriptions

Option	Description
ISO2 Country Code Sequence	<p>Determines the order in which the transform checks for phone data.</p> <p>Global: The transform uses the international regular expressions that are set in the Cleansing Package. If you keep Global, it should appear last in your list.</p> <p>Click the Browse button to open the Ordered Options Window. Choose specific ISO codes:</p> <ol style="list-style-type: none">1. Select an ISO code from the Available values list at left. Multiselect to choose more than one ISO code.2. Click Add. The software adds the ISO codes to the Selected Values list at right.3. If you kept Global, use the arrow buttons in the upper right to move global to the end of the list.4. As necessary, use the Remove and Remove All buttons.5. Click OK to close the window. <p>When you use the Ordered Options Window, be aware of the following information:</p> <ul style="list-style-type: none">• You can list a code only once in your sequence• The order in which you place the codes in the sequence determines the order in which Data Cleanse searches for phone information. Use the up and down arrow buttons to arrange the codes in the desired order.• If you want to keep Global in your list, move it last in your list of codes. For example, DE AU GLOBAL

❖ Example

If most of your data contains records from Germany and Australia, make sure that the ISO2 code for Germany comes first, followed by Australia: **DE | AU | GLOBAL**.

The transform first determines if the phone data matches phone data from Germany. If the data doesn't parse as German phone data, the transform then checks if the data matches Australia phone data. If the phone data doesn't parse as Australia phone data, the transform performs a global search and runs the data through the international regular expressions that are set in the Cleansing Package. Data Cleanse always performs a global search last, regardless of the order in which you place Global in the sequence.

Option	Description
North American Phone Parens Area	<p>Controls placement of parentheses () around the area code of phone number output following the North American Numbering Plan (NANP).</p> <p><i>Yes:</i> Includes the parentheses. For example, (123) 656-5000.</p> <p><i>No:</i> Omits the parentheses. For example, 123 656-5000.</p>
North American Phone Delimiter After Area	<p>Specifies placement of a delimiter between the area code and prefix phone output following the North American Numbering Plan (NANP).</p> <div> <p>i Note</p> <p>To use this option, set a delimiter in the North American Phone Delimiter option.</p> </div> <p><i>Yes:</i> Adds a delimiter. For example, 123-656-5000.</p> <p><i>No:</i> Excludes a delimiter. For example, 123 656-5000.</p>
North American Phone Delimiter	<p>Specifies a character to use as a delimiter for phone output following the North American Numbering Plan (NANP).</p> <div> <p>i Note</p> <p>To use this option, set the North American Phone Delimiter After Area to Yes.</p> </div> <p><i>Backslash (\):</i> Uses backward slashes as the delimiter in the phone number. For example, 123\656\5000.</p> <p><i>Dash (-):</i> Uses dashes as the delimiter in the phone number. For example, 123-656-5000.</p> <p><i>Slash (/):</i> Uses forward slashes as the delimiter in the phone number. For example, 123/656/5000.</p> <p><i>None:</i> Does not add a delimiter to the phone number. For example, 1236565000.</p> <p><i>Period (.):</i> Uses periods as the delimiter in the phone number. For example, 123.656.5000.</p> <p><i>Space:</i> Uses spaces as the delimiter in the phone number. For example, 123 656 5000.</p>
Phone Extension Text	Specifies the standard text for a phone extension. For example, "Ext."

Parent topic: [Data Cleanse options \[page 476\]](#)

Related Information

[Common options \[page 477\]](#)

[Report and Analysis \[page 478\]](#)

[Data Quality Statistics Settings \[page 479\]](#)

[Cleansing Package options \[page 482\]](#)

[Options \[page 485\]](#)
[Input word breaker \[page 486\]](#)
[Person standardization options \[page 487\]](#)
[Gender standardization options \[page 491\]](#)
[Firm standardization options \[page 492\]](#)
[Other standardization options \[page 493\]](#)
[Date options \[page 496\]](#)
[Parser configuration \[page 501\]](#)
[Memory cache \[page 502\]](#)

8.5.6.1.13 Parser configuration

Parser configuration options control which parsing engines Data Cleanse uses for parsing multiline fields and the order in which they are applied.

If a particular parser is not included, Data Cleanse does not look for that type of data in the input field.

Option	Description
Parser Sequence Multiline1-12	<p><i>Custom Parser</i>: Parses custom operational or product data for the category in the specified custom cleansing package.</p> <p><i>Date</i>: Parses data as a date.</p> <p><i>Email</i>: Parses data as an e-mail address.</p> <p><i>Firm</i>: Parses data as firm name.</p> <p><i>Person</i>: Parses data as a personal name</p> <p><i>Person or Firm</i>: Parses data as a personal or firm name.</p> <p><i>Phone</i>: Parses data as phone numbers. The transform parses any non-North American numbers first and can be output to the Phone and International_Phone fields. The transform parses North American numbers later using the North American Numbering Plan (NANP). The data can be output to Phone and North_American_Phone output fields. Define phone number patterns that do not follow the NANP in Cleansing Package Builder.</p> <p><i>SSN</i>: Parses data as U.S. Social Security number.</p> <p><i>UDPM</i>: Parses data using user-defined patterns created in Cleansing Package Builder</p>

Parent topic: [Data Cleanse options \[page 476\]](#)

Related Information

[Common options \[page 477\]](#)
[Report and Analysis \[page 478\]](#)

[Data Quality Statistics Settings \[page 479\]](#)

[Cleansing Package options \[page 482\]](#)

[Options \[page 485\]](#)

[Input word breaker \[page 486\]](#)

[Person standardization options \[page 487\]](#)

[Gender standardization options \[page 491\]](#)

[Firm standardization options \[page 492\]](#)

[Other standardization options \[page 493\]](#)

[Date options \[page 496\]](#)

[Phone Options group \[page 499\]](#)

[Memory cache \[page 502\]](#)

8.5.6.1.14 Memory cache

Memory cache allocates memory per thread for processing the Data Cleanse transform.

To improve performance, set the memory cache in kilobytes (KB) to allocate a certain amount of memory in the Data Cleanse transform. The option is *Memory in KB for Cache*. It is located in the *Options* group of the Data Cleanse transform.

❖ Example

If you have a 39 MB cleansing package, you can set the cache size to 40000 KB and use the rest of available memory for processing other transforms. Likewise, if your cleansing package is larger, you can increase the value.

Parent topic: [Data Cleanse options \[page 476\]](#)

Related Information

[Common options \[page 477\]](#)

[Report and Analysis \[page 478\]](#)

[Data Quality Statistics Settings \[page 479\]](#)

[Cleansing Package options \[page 482\]](#)

[Options \[page 485\]](#)

[Input word breaker \[page 486\]](#)

[Person standardization options \[page 487\]](#)

[Gender standardization options \[page 491\]](#)

[Firm standardization options \[page 492\]](#)

[Other standardization options \[page 493\]](#)

[Date options \[page 496\]](#)

[Phone Options group \[page 499\]](#)

8.5.6.2 Input fields for the Data Cleanse transform

Use the Data Cleanse input fields to map fields to use in the transform.

The following are recognized input fields that you can use in the input mapping for the Data Cleanse transform. The fields are listed alphabetically.

Input field name (Data Cleanse)	Description
Data_Source_ID	<p>Specifies the name of the data source that the record comes from.</p> <p>If you choose to generate data quality statistics tables, use this field as part of your mapping strategy, along with primary key fields and the output field Row_ID. This mapping identifies the source of individual records in any of the non summary data quality statistics tables, thus enabling you to trace problem data to the data source.</p> <p>This input field is applicable for the Data Cleanse transform, Global Address Cleanse transform, and Geocoder transform.</p>
Date1-6	Date. For example, 08/16/2004.
Email1-6	E-mail address.
Firm_Line1-6	Firm name, firm location, or both.
Firm_Location1-2	Location within a company or organization, such as a department, mail stop, room, or building.
Firm_Name1-2	Name of a company or organization.
Multiline1-12	Multiline data. The transform parses item types from this input in the order set in the Parser Sequence Multiline option, including parsers from custom cleansing packages.
Name_Line1-6	Whole name or names. May include job title.
Name_Or_Firm_Line1-6	Name of a person or organization.

Input field name (Data Cleanse)	Description
Option_Country (Dynamic input field)	<p>Option_Country</p> <p>Contains an ISO2 country code to help parse phone data and to determine the content domain sequence. Obtain ISO2 country code data either from an input file that contains the codes, or from an upstream transform such as Global Address Cleanse that is set up to output ISO2 codes.</p> <ul style="list-style-type: none"> Parse phone data: Data Cleanse uses the ISO2 country code to help determine the country code for parsing phone data when the Option_Country dynamic input field is populated with an ISO2 country code, and the country is included in the cleansing package. Map the Global Address Cleanse field ISO_Country_Code_2Char to the Option_Country input field. Set domain sequence: Data Cleanse uses the ISO2 code to determine content domain sequence when the Option_Country dynamic input field is populated with an ISO2 code. The transform automatically generates the output format based on the ISO2 country code and values in the Option_Language and Option_Region fields.
Option_Language (Dynamic input field)	Use these two fields with the Option_Country field to help determine the content domain and output format.
Option_Region (Dynamic input field)	<p>There are a few countries where Option_Language and Option_Region data is helpful to determine the most appropriate content domain and output format. The two fields are applicable only to certain countries. For example, Switzerland, Belgium, or Canada can be used for Option_Language and Option_Region.</p> <p>The transform uses these fields only to determine the most appropriate content domain and output format</p> <p>Map the Option_Country, Option_Language, and Option_Region input fields from the following Global Address Cleanse output fields in this order:</p> <ul style="list-style-type: none"> ISO_Country_Code_2Char Language Region1 <p>These three input field work together to determine the Option_Content_Domain_Sequence and-or Option_Output_Format. For more information and examples, see "About Domains" in the <i>Designer Guide</i>.</p>
Option_Content_Domain_Sequence (Dynamic input field)	The content domain sequence. The valid predefined values are: AR, CS, DA, DE, EN_AU, EN_GB, EN_IN, EN_US, ES_MX, ES_ES, FR, HU, ID, IT, JA, MS, NL, NO, PL, PT_BR, PT_PT, RO, RU, SK, SV, ZH, and GLOBAL.
Option_Output_Format (Dynamic input field)	<p>The format for output specified as an abbreviation of the domain.</p> <p>The valid predefined values are: AR, CS, DA, DE, EN_AU, EN_GB, EN_IN, EN_US, ES_MX, ES_ES, FR, HU, ID, IT, JA, MS, NL, NO, PL, PT_BR, PT_PT, RO, RU, SK, SV, and ZH.</p>
Person1_Family_Name1 Person2_Family_Name1	Discrete family name (for example, Smith).

Input field name (Data Cleanse)	Description
Person1_Family_Name2	Second discrete family name.
Person2_Family_Name2	May be useful for cultures where people are known by both paternal and maternal family names. If your input data contains two family name fields, map the first to Person1_Family_Name1 and the second to Person1_Family_Name2.
Person1_Given_Name1-2	Discrete given names (for example, John or B.).
Person2_Given_Name1-2	
Person1_Honorary_Postname	Honorary postname indicating certification, academic degree, or affiliation, such as CPA.
Person2_Honorary_Postname	
Person1_Maturity_Postname	Maturity postname indicating heritage, such as Jr., Sr., III.
Person2_Maturity_Postname	
Person1_Prenome	Discrete prename, such as Mr., Mrs., Dr., or Lt. Col.
Person2_Prenome	
Person1_Title	Discrete job title, such as Software Engineer.
Person2_Title	
Phone1-6	Phone number. Data Cleanse first tries to parse the number as an international phone number. If that fails, it tries to parse it as a North American phone number.
SSN1-6	U.S. Social Security number.
Title_Line1-6	Job title (for example, Accountant).
UDPM1-4	Input field associated with patterns and rules defined in the user-defined type of Reference Data in Cleansing Package Builder. For example, CN244-56.

Parent topic: [Data Cleanse \[page 474\]](#)

Related Information

[Data Cleanse options \[page 476\]](#)

[Output fields for the Data Cleanse transform \[page 505\]](#)

[Dynamic transform settings \[page 354\]](#)

8.5.6.3 Output fields for the Data Cleanse transform

Use the output fields to map output from the Data Cleanse transform.

The following are recognized output fields that you can use in the output mapping for the Data Cleanse transform. By default, the Extra, Info_Code, and Status_Code fields are always available, additional output

fields are displayed based on the mapped input fields and the selected parser sequence multiline options. Additionally, the Ignore field may be available.

You can use the [Filter Output Fields](#) option to display a complete list of output fields. The fields are listed alphabetically.

Output field name (Data Cleanse)	Content type	Description
Date	Date	A date that is parsed.
Date_Day	Date	The day that is parsed from the date.
Date_Month	Date	The month that is parsed from the date.
Date_Year	Date	The year that is parsed from the date.
Dual_Name	None	Set of components resulting from one input field that contains two names separated by a connecting word such as "and" or "or."
<div> <div>❖ Example</div> <div>Input Terry and Kris Johnson Output Terry Johnson and Kris Johnson</div> </div>		
<div> <div>❖ Example</div> <div>Input Terry Johnson or Kris Adams Output Terry Johnson or Kris Adams</div> </div>		
Email	Email	An entire e-mail address.
Email_Domain_All	Email	The domain of the e-mail address. For example, sap.com.
Email_Domain_Fifth	Email	In an e-mail address with more than one domain listed, this field parses the fifth to last domain.
Email_Domain_Fourth	Email	In an e-mail address with more than one domain listed, this field parses the fourth to last domain.
Email_Domain_Host	Email	The host of the e-mail address (the first item listed after the @ symbol). For example, in "joex@sap.com", "sap" is returned.
Email_Domain_Second	Email	In an e-mail address with more than one domain listed, this field parses the second to last domain.
Email_Domain_Third	Email	In an e-mail address with more than one domain listed, this field parses the third to last domain.
Email_Domain_Top	Email	The last listed domain of the e-mail address. For example, .com.
Email_Is_ISP	Email	The e-mail address is a known ISP (internet service provider) or e-mail domain name listed in the e-mail data type of Reference Data in Cleansing Package Builder.
Email_User	Email	The user name of the e-mail address. For example, in "joex@sap.com", "joex" is returned.

Output field name (Data Cleanse)	Content type	Description
Extra	None	Any data that is not parsed by any of the active parsers and thus Data Cleanse does not recognize the data as fitting one of the other output fields.
Family_Name1	Family_Name	Family name (for example, Smith).
Family_Name1_Match_Std1-6	Family_Name1_Match_Std	<p>The match standard for family names.</p> <p>This field is only used with cleansing packages that include name data in more than one script. The match standards include the name as it is written in alternate script types. For example, for a family name included in the Japanese dictionary in Kanji script, the match standards include Kana renditions of the name.</p> <p>If the dictionary does not have an alias entry, the output field is empty.</p> <p>Compare with Match_Family_Name.</p>
Family_Name2	Family_Name 2	Second family name. May be used to output the paternal and maternal family names to separate fields.
Family_Name2_Match_Std1-6	Family_Name2_Match_Std	<p>The match standard for second family names.</p> <p>This field is only used with cleansing packages that include name data in more than one script. The match standards include the name as it is written in alternate script types. For example, for a family name included in the Japanese dictionary in Kanji script, the match standards include Kana renditions of the name.</p> <p>If the dictionary does not have an alias entry, the output field is empty.</p> <p>Compare with Match_Family_Name.</p>
Family_Name_Full	Family_Name_Full	<p>Combines Family_Name1 and Family_Name2 into one field.</p> <div> <p>♣ Example</p> <p>Family_Name1 = Smith Family_Name2 = Jones Family_Name_Full = Smith Jones</p> </div>
Firm	Firm	The name of a company or organization.
Firm_Location	Firm_Location	A location within a company or organization, such as a department. For example, Mailstop.

Output field name (Data Cleanse)	Content type	Description
Firm_Match_Std1-6	Firm_Match_Std	<p>The match standard for firms. For example, HP is the match standard or alias for Hewlett Packard.</p> <p>If the cleansing package does not have an alias entry, the output field is empty.</p> <p>Compare with Match_Firm.</p>
Firm_Location_Match_Std1-6	Firm_Location_Match_Std	<p>The match standard for firm locations. For example, MS is the match standard or alias for mailstop.</p> <p>If the dictionary does not have an alias entry, the output field is empty.</p>
Gender	None	<p>The gender description. The following output is available:</p> <p><i>Ambiguous</i>: The name does not reliably indicate a gender. The name could be either male or female. For example, Pat.</p> <p><i>Male_Strong</i>: High confidence that the person is male. That is, the name belongs to someone who is almost certainly a male. For example, John.</p> <p><i>Male_Weak</i>: Some confidence that the person is male. That is, the name belongs to someone who is probably male. For example, Terry.</p> <p><i>Female_Strong</i>: High confidence that the person is female. That is, the name belongs to someone who is almost certainly a female. For example, Mary.</p> <p><i>Female_Weak</i>: Some confidence that the person is female. That is, the name belongs to someone who is probably a female. For example, Lynn.</p> <p>For dual names, the following output is also available:</p> <p><i>Multi_Names_Ambiguous</i>: At least one of the names does not reliably indicate a gender. For example, Pat and John.</p> <p><i>Multi_Names_Female</i>: Some or high confidence that both of the names belong to people who are female. For example, Mary and Lynn.</p> <p><i>Multi_Names_Male</i>: Some or high confidence that both of the names belong to people who are male. For example, John and Terry.</p> <p><i>Multi_Names_Mixed</i>: Some or high confidence that one of the names belongs to a person who is female, and the other name belongs to a person who is male. For example, Lynn and John.</p>

Output field name (Data Cleanse)	Content type	Description
Gender_ID	None	<p>A numeric value that corresponds to the gender description</p> <p>0:" Unassigned</p> <p>1: Male_Strong</p> <p>2: Male_Weak</p> <p>3: Ambiguous</p> <p>4: Female_Weak</p> <p>5: Female_Strong</p> <p>6: Multi_Names_Mixed</p> <p>7: Multi_Names_Male</p> <p>8: Multi_Names_Female</p> <p>9: Multi_Names_Ambiguous</p>
Given_Name1	Given_Name	Given name (for example, Robert).
Given_Name1_Match_Std1-6	Given_Name1_Match_Std	<p>The match standard for given names. For example, the application can tell you that Patrick and Patricia are potential matches for the given name Pat.</p> <p>Match standards can help you overcome two types of matching problems alternate spellings (Catherine and Katherine) and nicknames (Pat and Patty).</p> <p>Compare with Match_Given_Name1.</p>
Given_Name2	Given_Name2	Second given name.
Given_Name2_Match_Std1-6	Given_Name2_Match_Std	<p>The match standard for second given names. For example, the application can tell you that Patrick and Patricia are potential matches for the given name Pat.</p> <p>Compare with Match_Given_Name2.</p>
Given_Name_Full	Given_Name_Full	Combines Given_Name1 and Given_Name2 into one field.
<div> <div>♣ Example</div> <div> Given_Name1 = Jonathan Given_Name2 = Peter Given_Name_Full = Jonathan Peter </div> </div>		
Honorary_Postname	Postname	Honorary postname indicating certification, academic degree, or affiliation. For example, CPA.

Output field name (Data Cleanse)	Content type	Description
Honorary_Postname_Match_Std1-6	Postname_Match_Std	<p>The match standard for an honorary postname. For example, M.B.A. is the match standard or alias for MBA.</p> <p>If the dictionary does not have an alias entry, the output field is empty.</p>
Info_Code	None	The code that identifies the rows that may require manual review because the data is suspect. See Detailed descriptions of information codes [page 829] for a list of information codes and descriptions.
International_Phone	Phone	The entire international phone number, including extra items such as the country code.
International_Phone_Country_Code	Phone	The country code of an international phone number.
International_Phone_Country_Name	Phone	The name of the country of origin of an international phone number.
International_Phone_Line	Phone	The portion of the international phone number that is not the country code or the city code.
International_Phone_Locality_Code	Phone	The locality code of an international phone number.
Match_Family_Name	Family_Name1	<p>The combined standardized form of FamilyName1 and FamilyName2 with a space between used in the Match transform during the comparison process. The transform outputs data in uppercase, removes apostrophes, and replaces other punctuation with a single space. Removes PreFamilyName data.</p> <p>Compare with Family_Name1_Match_Std1-6 and Family_Name2_Match_Std1-6.</p>
Match_Firm	Firm	<p>A form of Firm that may be used in the Match transform during the comparison process. The transform outputs data in uppercase, removes apostrophes, and replaces other punctuation with a single space, and removes data that is extraneous for matching purposes. This extraneous data includes business types such as Ltd. and GmbH, and noise words such as The, And, and Of.</p> <div> <p>i Note</p> <p>Some words are classified to be removed from all domains, while others are language-specific and are classified to be removed in specific cultural domains.</p> </div> <p>Compare with Firm_Match_Std1-6.</p>

Output field name (Data Cleanse)	Content type	Description
Match_Given_Name1	Given_Name1	<p>The standardized form of GivenName1 used in the Match transform during the comparison process. The transform outputs data in uppercase, removes apostrophes, and replaces other punctuation with a single space. Removes PreGivenName data.</p> <p>Compare with Given_Name1_Match_Std1-6.</p>
Match_Given_Name2	Given_Name2	<p>The standardized form of GivenName2 used in the Match transform during the comparison process. The transform outputs data in uppercase, removes apostrophes, and replaces other punctuation with a single space. Removes PreGivenName data.</p> <p>Compare with Given_Name2_Match_Std1-6.</p>
Match_Maturity_Postname	Postname	<p>The standardized form of MaturityPostname used in the Match transform during the comparison process. The transform outputs data in uppercase, removes apostrophes, and replaces other punctuation with a single space.</p> <p>Compare with Maturity_Postname_Match_Std1-6.</p>
Match_Person	Person	<p>A form of Person that may be used in the Match transform during the comparison process. The transform outputs data in uppercase, removes apostrophes, and replaces other punctuation with a single space, and removes data that is extraneous for matching purposes. Extraneous data includes pre-given name, pre-family name, and pre-name as well as honorary and maturity post names and name designators.</p>
Match_Phone	Phone	<p>The form of Phone used in the Match transform during the comparison process. Data is output as a string of digits. Spaces, punctuation, alphabetical characters, and leading zeros are removed.</p>
Match_Prename	Prename	<p>The standardized form of Prename used in the Match transform during the comparison process. The transform outputs data in uppercase, removes apostrophes, and replaces other punctuation with a single space.</p> <p>Compare with Prename_Match_Std1-6.</p>
Maturity_Postname	Postname	<p>Maturity postname indicating heritage, such as Jr., Sr., III.</p>
Maturity_Postname_Match_Std1-	Postname_Match_Std	<p>The match standard for a maturity postname. For example, Sr. is a match standard or alias for Senior.</p> <p>If the cleansing package does not have an alias entry, the output field is empty.</p>
Name_Connector	None	<p>The connector component of a dual name. For example, and.</p>

Output field name (Data Cleanse)	Content type	Description
Name_Designator	None	Name designator such as Attn or c/o.
Name_Special	None	Term that generically describes a person. For example, occupant or current resident.
North_American_Phone	Phone	An entire North American Numbering Plan (NANP) phone number.
North_American_Phone_Area_Code	Phone	The area code parsed from the phone number.
North_American_Phone_Extension	Phone	An extension parsed from the phone number.
North_American_Phone_Line	Phone	The last four numbers (excluding an extension) parsed from a phone number. In (123) 456-7890, 7890 is returned.
North_American_Phone_Prefix	Phone	The middle three numbers parsed from a phone number. In (123) 456-7890, 456 is returned.
North_American_Phone_Type	Phone	The type of phone number that was parsed, if it is included with the input. For example, Home or Work.
Person	None	Set of components that define a single person. For example, Thomas Williams-Doyle Sr., M.D.
Phone	Phone	Shows the phone number that was identified as either North American or International.
<div> i Note You have the option to standardize North American phone numbers. However, Data Cleanse does not standardize or enhance international phone numbers, and outputs the number as it was input. </div>		
Prenome	Prenome	Prenome (for example, Mr.).
Prenome_Match_Std1-6	Prenome_Match_Std	The match standard for a prename. For example, Mr. is the match standard or alias for Mister. If the cleansing package does not have an alias entry, the output field is empty. Compare with Match_Prenome.

Output field name (Data Cleanse)	Content type	Description
Row_ID	Statistics	<p>An internal ID generated by the transform that uniquely identifies a row processed by that transform.</p> <p>Use this field to join the input and output to the applicable non-summary data quality statistics table.</p> <p>The software issues a warning if you select to generate any data quality statistics tables in the transform, and you don't include the Row_ID output field. The software also issues a warning if you map this output field and don't select to generate any non-summary data cleanse statistics tables in the transform.</p> <p>Applicable transforms are Data Cleanse, Global Address Cleanse, and Geocoder.</p>
Rule_Label	None	Retrieves the rule that parsed the indicated item.
Score	None	Retrieves the confidence score for a parsed item.
SSN	SSN	The entire Social Security number.
SSN_Area	SSN	The first three numbers of the Social Security number.
SSN_Group	SSN	The fourth and fifth numbers within a Social Security number.
SSN_Serial	SSN	The last four numbers in a Social Security number.
Status_Code	None	The code describing how the data was standardized when the job is set for standardization, or how the data could be standardized when the job settings are not set for standardization.
Title	Title	Job or occupational title of a person. For example, Manager.
Title_Match_Std1-6	Title_Match_Std	<p>The match standard for title. For example, CFO is the match standard or alias for Chief Financial Officer.</p> <p>If the dictionary does not have an alias entry, the output field is empty.</p>
UDPM	None	Attribute field defined in User-defined pattern rules in Cleansing Package Builder Reference Data.
UDPM_Subcomponent1-5	None	Subcomponents of the UDPM attribute field defined in a User-defined pattern rule.

Parent topic: [Data Cleanse \[page 474\]](#)


Related Information

[Data Cleanse options \[page 476\]](#)
[Input fields for the Data Cleanse transform \[page 503\]](#)

8.5.7 DSF2® Walk Sequencer

Use the DSF2 Walk Sequencer transform to add walk sequencing information to your output data.

DSF2 Walk Sequencer information

Characteristic	Description
	DSF2 Walk Sequencer icon
Use	<p>Use the DSF2 Walk Sequencer transform to enhance your data with information that qualifies your mailings for United States Postal Service (USPS) walk sequence discounts. Potential discounts include:</p> <ul style="list-style-type: none">• Carrier Route• Walk Sequence• 90% Residential Saturation• 75% Total Active Saturation <p>DSF2 walk sequencing is called “pseudo” sequencing because it mimics USPS walk sequencing. Where USPS walk-sequence numbers cover every address, DSF2 walk sequence processing provides pseudo sequence numbers for the addresses in the particular database file only.</p> <p>The software uses DSF2 data to assign sequence numbers for all addresses that are delivery point validation (DPV) confirmed delivery points.</p> <p>The transform outputs addresses that are not DPV confirmed delivery points as blank or with 0000 in the Walk_Sequence_Number output field.</p> <div><p>i Note</p><p>Before processing your data with the DSF2 Walk Sequencer transform, process it through CASS-certified software such as the USA Regulatory Address Cleanse transform.</p></div>

Characteristic	Description
Content objects	Use Data Quality blueprints and other content objects, including several blueprints to run DSF2 certifications, to help set up your jobs. These blueprints are located in <code><LINK_DIR>\DataQuality\Certifications</code> .

[Common options \[page 477\]](#)

There is one common option in the Data Cleanse transform: *Run as Separate Process*.

[Reference Files \[page 517\]](#)

The Delivery Statistics file contains the reference information for DSF2 walk sequencing.

[Processing Options \[page 517\]](#)

Enter job processing site and certification test mode information.

[USPS License Information \[page 518\]](#)

USPS license information such as license ID and licensee name is required for DSF2 walk sequencing.

[Data Collection Config \[page 519\]](#)

Set options to optimize your data flow by controlling break key formation based on values such as postcode and sort code route fields.

[Input fields for the DSF2 Walk Sequencer transform \[page 520\]](#)

Use input fields to map your source data to use in the transform.

[Output fields for the DSF2 Walk Sequencer transform \[page 522\]](#)

The transform outputs walk sequence number information the DSF2 Walk Sequencer output fields.

Parent topic: [Data Quality transforms \[page 437\]](#)

Related Information

[Blueprints and other content objects for download \[page 438\]](#)

[About Data Quality fields \[page 445\]](#)

[About data quality statistics \[page 448\]](#)

[Associate \[page 454\]](#)

[Country ID \[page 472\]](#)

[Data Cleanse \[page 474\]](#)

[Geocoder \[page 524\]](#)

[Global Address Cleanse \[page 552\]](#)

[Global Suggestion List \[page 640\]](#)

[Match \[page 651\]](#)

[USA Regulatory Address Cleanse \[page 710\]](#)

[Address Cleanse reference \[page 765\]](#)

[Data Cleanse reference \[page 823\]](#)

[Common options \[page 477\]](#)

8.5.7.1 Common options

There is one common option in the Data Cleanse transform: [Run as Separate Process](#).

Option	Description
Run As Separate Process	Yes: Splits the transform into a separate process. No: Keeps the transform in the same process as the rest of the data flow.

Parent topic: [Data Cleanse options \[page 476\]](#)

Parent topic: [DSF2® Walk Sequencer \[page 514\]](#)

Related Information

[Report and Analysis \[page 478\]](#)
[Data Quality Statistics Settings \[page 479\]](#)
[Cleansing Package options \[page 482\]](#)
[Options \[page 485\]](#)
[Input word breaker \[page 486\]](#)
[Person standardization options \[page 487\]](#)
[Gender standardization options \[page 491\]](#)
[Firm standardization options \[page 492\]](#)
[Other standardization options \[page 493\]](#)
[Date options \[page 496\]](#)
[Phone Options group \[page 499\]](#)
[Parser configuration \[page 501\]](#)
[Memory cache \[page 502\]](#)
[Reference Files \[page 517\]](#)
[Processing Options \[page 517\]](#)
[USPS License Information \[page 518\]](#)
[Data Collection Config \[page 519\]](#)
[Input fields for the DSF2 Walk Sequencer transform \[page 520\]](#)
[Output fields for the DSF2 Walk Sequencer transform \[page 522\]](#)

8.5.7.2 Reference Files

The Delivery Statistics file contains the reference information for DSF2 walk sequencing.

Option	Description
Delstats Directory	<p>Type or browse to the path for the Delivery Statistics file, <code>dsf.dir</code>.</p> <p>Best practice is to use a substitution variable for the directory location. Data Services provides the substitution variable <code><\$\$RefFilesAddressCleanse></code> for your use.</p> <p>The Delivery Statistics directory file provides counts of business and residential addresses per ZIP Code and per carrier route.</p> <p>Data Services provides the Delivery Statistics file with the U.S. National Directory delivery on a periodic basis.</p>

For the most up to date information about all of the directories, see the directory update letters for SAP Data Services address cleansing and geocoding at https://uacp2.hana.ondemand.com/viewer/p/ADDRESSING_DIRECTORIES.

Parent topic: [DSF2® Walk Sequencer \[page 514\]](#)

Related Information

[Common options \[page 477\]](#)

[Processing Options \[page 517\]](#)

[USPS License Information \[page 518\]](#)

[Data Collection Config \[page 519\]](#)

[Input fields for the DSF2 Walk Sequencer transform \[page 520\]](#)

[Output fields for the DSF2 Walk Sequencer transform \[page 522\]](#)

8.5.7.3 Processing Options

Enter job processing site and certification test mode information.

Option	Description
Site Location	<p>The name of your company location at which this job is processed.</p> <p>If your company has multiple locations that provide DSF2 walk sequence processing, ensure that you indicate the location where you process the job that you are currently configuring.</p> <p>Best practice is to use a substitution variable for this value. Data Services provides the substitution variable <code><\$\$DSF2SiteLocation></code> for your use.</p>

Option	Description
<i>USPS Certification Testing Mode</i>	<p>The type of certification you are performing for DSF2 walk sequence if applicable. Options include:</p> <p><i>None</i>: Not performing any DSF2 walk sequence certifications. This is the default setting. Choose this option if you are processing a job that will not be submitted to the USPS for DSF2 certification.</p> <p><i>Invoice</i>: Performing certification for DSF2 walk sequence invoice certification. For invoice certification, you certify that the software assigns walk sequence numbers correctly and creates the Delivery Sequence Invoice report.</p> <p><i>Sequence</i>: Performing certification for DSF2 sequence certification. For sequence certification, you certify that the software assigns walk sequence numbers correctly, creates the Delivery Sequence Invoice report, and creates the SEQ log file.</p>

Parent topic: [DSF2® Walk Sequencer \[page 514\]](#)

Related Information

[Common options \[page 477\]](#)

[Reference Files \[page 517\]](#)

[USPS License Information \[page 518\]](#)

[Data Collection Config \[page 519\]](#)

[Input fields for the DSF2 Walk Sequencer transform \[page 520\]](#)

[Output fields for the DSF2 Walk Sequencer transform \[page 522\]](#)

8.5.7.4 USPS License Information

USPS license information such as license ID and licensee name is required for DSF2 walk sequencing.

Option	Description
<i>DSF2 License ID</i>	Enter your DSF2 identification number as the USPS assigned it to you. Best practice is to use a substitution variable. Data Services provides the substitution variable <code><\$\$DSF2LicenseeID></code> for your use.
<i>Licensee Name</i>	Enter the name of the DSF2 licensed service provider. Best practice is to use a substitution variable. Data Services provides the substitution variable <code><\$USPSLicenseeName></code> for your use.
<i>List ID</i>	Enter the unique 6-digit identification code that you as the DSF2 licensee assigned to the owner of the mailing list being processed.

Parent topic: [DSF2® Walk Sequencer \[page 514\]](#)

Related Information

[Common options \[page 477\]](#)

[Reference Files \[page 517\]](#)

[Processing Options \[page 517\]](#)

[Data Collection Config \[page 519\]](#)

[Input fields for the DSF2 Walk Sequencer transform \[page 520\]](#)

[Output fields for the DSF2 Walk Sequencer transform \[page 522\]](#)

8.5.7.5 Data Collection Config

Set options to optimize your data flow by controlling break key formation based on values such as postcode and sort code route fields.

The settings in the Data Collection Config group control the break key formation. Break keys optimize your data flow by sorting data to form collections of input records that have the same `<Postcode1>` and `<Sortcode_Route>` field values.

Option	Description
Replace Null With Space	<p>Specifies whether to convert NULL values to blank spaces in the break key fields. Eliminating NULL values helps to standardize data in the field. With standardized data, the transform can properly form break groups that are consistent in size.</p> <p>Yes: Convert NULL to blank spaces.</p> <p>No: Do not convert NULL to blank spaces.</p>
Right Pad With Spaces	<p>Specifies whether to right pad the break key field with spaces.</p> <p>Because the break key is used for sorting and aggregating, it is sensitive to the position in which data is placed. When you enable right-padding for the break key fields, the transform can properly form break groups that are consistent in size.</p> <p>Yes: Right-pad fields with blank spaces.</p> <p>No: Do not right-pad fields with blank spaces.</p>

→ Tip

If the [Replace NULL with space](#) and the [Right pad with spaces](#) options are set to **Yes**, then fields with NULL values are replaced with all spaces on the right, to the length of the field.

Option	Description
<i>Presorted Data</i>	<p>Specifies whether the input data has been presorted or not. To make your input data more consistent, it is best to have the software sort data by the break key fields (Postcode1 and Sortcode_Route).</p> <p>Yes: The input data has already been presorted by Postcode1 and Sortcode_Route.</p> <p>No: The input data has not been sorted yet.</p> <div> <p>→ Tip</p> <p>Choosing <i>No</i> allows the software to sort your data by Postcode1 and Sortcode_Route. Best practice is to set this option to <i>No</i>.</p> </div>

Parent topic: [DSF2® Walk Sequencer \[page 514\]](#)

Related Information

[Common options \[page 477\]](#)

[Reference Files \[page 517\]](#)

[Processing Options \[page 517\]](#)

[USPS License Information \[page 518\]](#)

[Input fields for the DSF2 Walk Sequencer transform \[page 520\]](#)

[Output fields for the DSF2 Walk Sequencer transform \[page 522\]](#)

8.5.7.6 Input fields for the DSF2 Walk Sequencer transform

Use input fields to map your source data to use in the transform.

The following table describes the recognized input fields to use with the DSF2 Walk Sequencer transform.

Input field name (DSF2 Walk Sequencer)	Description
Delivery_Point	The two-digit DPBC code.

Input field name (DSF2 Walk Sequencer)	Description
DPV_Status	<p>The DPV status component that is generated for this record.</p> <p><i>D</i>: The primary range is a confirmed delivery point, but the secondary range was not available on input.</p> <p><i>L</i>: The address triggered DPV locking.</p> <p><i>N</i>: The address is not a valid delivery point. The <code><Walk_Sequence_Number></code> output field is 0000.</p> <p><i>S</i>: The primary range is a valid delivery point, but the parsed secondary range is not valid in the DPV directory.</p> <p><i>Y</i>: The address is a confirmed delivery point. The primary range and secondary range (if present) are valid.</p> <p><code><blank></code>: A blank output value indicates that <i>Enable DPV</i> is set to <i>No</i>, DPV processing is locked, or the transform cannot assign the input address. The <code><Walk_Sequence_Number></code> output field is blank.</p>
DSF2_Business_Indicator (optional)	<p>Residential or business indicator. Use this information to lower your parcel shipping costs. Some parcel delivery services charge more for delivery to residential addresses.</p> <p><i>Y</i>: Business address.</p> <p><i>N</i>: Not a business address.</p> <p><code><blank></code>: Transform did not look up address.</p>
LOT	The Line-of-Travel number.
LOT_Order	<p>The Line-of-Travel sortation:</p> <p><i>A</i>: Ascending</p> <p><i>D</i>: Descending</p> <p>LOT codes are required for non automated, CART presorting in Standard Mail, Enhanced Carrier Route Subclass.</p>
Postcode1	The five-digit primary ZIP Code. It does not include the four digit ZIP4 Code.
Postcode2	The four-digit ZIP4 code. On a mail piece, this code follows the primary postal code with a hyphen placed between, for example, 54601-1234.
Sortcode_Route	The four-digit carrier route number.

Parent topic: [DSF2® Walk Sequencer \[page 514\]](#)

Related Information

[Common options \[page 477\]](#)

[Reference Files \[page 517\]](#)

[Processing Options \[page 517\]](#)
[USPS License Information \[page 518\]](#)
[Data Collection Config \[page 519\]](#)
[Output fields for the DSF2 Walk Sequencer transform \[page 522\]](#)

8.5.7.7 Output fields for the DSF2 Walk Sequencer transform

The transform outputs walk sequence number information the DSF2 Walk Sequencer output fields.
The following table contains descriptions for the recognized DSF2 Walk Sequencer output fields.

Output field name (DSF2 Walk Sequencer)	Description
Active_Del_Discount	<p>Indicates whether the Postcode1—Sortcode route combination qualifies for the 75% total active delivery discount.</p> <p><i>Y</i>: The Postcode1—Sortcode route combination qualifies for the 75% total active delivery discount.</p> <p><i>N</i>: The Postcode1—Sortcode route combination does not qualify for the 75% total active delivery discount.</p> <p><i><blank></i>: The record was not sequenced.</p> <div><div>→ Tip</div><p>Active deliveries include residential, business, and PO Box addresses.</p></div>
Residential_Sat_Discount	<p>Indicates whether the Postcode1—Sortcode route combination qualifies for the 90% residential saturation discount.</p> <p><i>Y</i>: The Postcode1—Sortcode route combination qualifies for the 90% residential saturation discount.</p> <p><i>N</i>: The Postcode1—Sortcode route combination does not qualify for the 90% residential saturation discount.</p> <p><i><blank></i>: The record was not sequenced.</p>

Output field name (DSF2 Walk Sequencer)	Description
Sortcode_Route_Discount	<p>Indicates whether the Postcode1—Sortcode route combination qualifies for the Sortcode (Carrier Route) discount.</p> <p><i>Y</i>: The Postcode1—Sortcode route combination qualifies for the Sortcode discount.</p> <p><i>N</i>: The Postcode1—Sortcode route combination does not qualify for the Sortcode discount.</p> <p><i><blank></i>: The record was not sequenced.</p> <div> <p>→ Tip</p> <p>Mailers must have 10 or more deliveries to the same Postcode1—Sortcode combination to qualify for the discount.</p> </div>
Walk_Sequence_Discount	<p>Indicates whether the Postcode1—Sortcode route combination qualifies for the walk sequence discount.</p> <p><i>Y</i>: The Postcode1—Sortcode route combination qualifies for the walk sequence discount.</p> <p><i>N</i>: The Postcode1—Sortcode route combination does not qualify for the walk sequence discount.</p> <p><i><blank></i>: The record was not sequenced.</p> <div> <p>→ Tip</p> <p>Mailers must have 125 or more sequenced delivery points for each Postcode1—Sortcode route combination to qualify for the discount.</p> </div>
Walk_Sequence_Number	<p>Indicates the sequence number from 0000 to 9999.</p> <div> <p>i Note</p> <p>If the <i><Postcode2></i> field is blank, this field is blank.</p> </div> <div> <p>i Note</p> <p>If the <i><Postcode2></i> field is not blank and <i>DPV_Status</i> is <i>N</i>, then the field contains 0000.</p> </div>

Parent topic: [DSF2® Walk Sequencer \[page 514\]](#)

Related Information

[Common options \[page 477\]](#)

[Reference Files \[page 517\]](#)


[Processing Options \[page 517\]](#)

[USPS License Information \[page 518\]](#)
[Data Collection Config \[page 519\]](#)
[Input fields for the DSF2 Walk Sequencer transform \[page 520\]](#)


8.5.8 Geocoder

Use the Geocoder transform to append latitude, longitude, and US census data to your data.

Geocoder information

Characteristic	Description
	Geocoder icon

Characteristic	Description
Use	<p>The Geocoder transform uses geographic coordinates expressed as latitude and longitude, addresses, and point-of-interest (POI) data. Append addresses, latitude and longitude, US census data, and other information to your data.</p> <div> <p>i Note</p> <p>Data Services uses the World Geodetic System (WGS84) for latitude and longitude results. WGS84 is defined and maintained by the US National Geospatial-Intelligence Agency (NGA).</p> </div> <p>The transform has options to enable generation of data quality statistics tables in the repository for address geocoding mode.</p> <div> <p>i Note</p> <p>Not applicable for reverse geocoding and point of interest (POI) textual search.</p> </div> <p>These statistics provide insight into how the software cleansed and assigned your data by generating aggregated summary statistics and specific record-level statistics.</p> <p>The Geocoder transform has three modes of geocode processing:</p> <ul style="list-style-type: none"> • Address geocoding: The Geocoder transform assigns geographic data. Based on the completeness of the input address data, the transform returns multiple levels of latitude and longitude data. Appending different levels of latitude and longitude information to your data may help your organization to target certain population sizes and other regional geographical data. • Reverse geocoding: The Geocoder transform identifies the closest address or point of interest based on an input reference location. • POI textual search: The Geocoder transform uses address fields and POI name or type fields as search criteria to match with points of interest. The transform outputs the results in the <code><Result_List XML></code> output field. <p>Best practice is to use the Geocoder transform with the Global Address Cleanse or USA Regulatory Address Cleanse transform.</p>

Characteristic	Description
Content objects	Download blueprints and other content objects for Geocoder from the SAP Community WIKI at https://wiki.scn.sap.com/wiki/display/EIM/SAP+Data+Services+Blueprints  .

Parent topic: [Data Quality transforms \[page 437\]](#)

Related Information

[Blueprints and other content objects for download \[page 438\]](#)

[About Data Quality fields \[page 445\]](#)

[About data quality statistics \[page 448\]](#)

[Associate \[page 454\]](#)

[Country ID \[page 472\]](#)

[Data Cleanse \[page 474\]](#)

[DSF2® Walk Sequencer \[page 514\]](#)

[Global Address Cleanse \[page 552\]](#)

[Global Suggestion List \[page 640\]](#)

[Match \[page 651\]](#)

[USA Regulatory Address Cleanse \[page 710\]](#)

[Address Cleanse reference \[page 765\]](#)

[Data Cleanse reference \[page 823\]](#)

[Data Quality Statistics Settings \[page 479\]](#)

8.5.8.1 Common options

There is one common option in the Data Cleanse transform: *Run as Separate Process*.

Option	Description
Run As Separate Process	<p>Yes: Splits the transform into a separate process.</p> <p>No: Keeps the transform in the same process as the rest of the data flow.</p>

Parent topic: [Data Cleanse options \[page 476\]](#)

Parent topic: [DSF2® Walk Sequencer \[page 514\]](#)

Related Information

[Report and Analysis \[page 478\]](#)
[Data Quality Statistics Settings \[page 479\]](#)
[Cleansing Package options \[page 482\]](#)
[Options \[page 485\]](#)
[Input word breaker \[page 486\]](#)
[Person standardization options \[page 487\]](#)
[Gender standardization options \[page 491\]](#)
[Firm standardization options \[page 492\]](#)
[Other standardization options \[page 493\]](#)
[Date options \[page 496\]](#)
[Phone Options group \[page 499\]](#)
[Parser configuration \[page 501\]](#)
[Memory cache \[page 502\]](#)
[Reference Files \[page 517\]](#)
[Processing Options \[page 517\]](#)
[USPS License Information \[page 518\]](#)
[Data Collection Config \[page 519\]](#)
[Input fields for the DSF2 Walk Sequencer transform \[page 520\]](#)
[Output fields for the DSF2 Walk Sequencer transform \[page 522\]](#)

8.5.8.2 Report and Analysis

Use this option to generate report data for the Geocoder transform.

Option	Description
Generate Report Data	<p>Specifies whether to generate report data for this transform.</p> <p>Yes: Generates report data for this transform.</p> <p>No: Turns off report data generation. If you do not need to generate reports, for example, during testing, set this option to No to improve performance.</p> <div><p>Note</p><p>This option is available only for the address geocoding mode. If you set this value to Yes for the reverse geocoding and POI textual search modes, the software generates a warning and does not generate a report.</p></div>

8.5.8.3 Data Quality Statistics Settings

The Data Quality Statistics Settings control the generation of data quality statistics tables.

The [Data Quality Statistics Settings](#) options in the following table apply to either Data Cleanse, Global Address Cleanse, or both transforms. Some also apply to the Geocoder transform. The following table includes a column that indicates the applicable transforms.

Data quality statistics option descriptions

Option	Transform	Description
Generate Cleanse Statistics Table	Data Cleanse	<p>Specifies to generate the Cleanse Statistics table. Options include:</p> <ul style="list-style-type: none">Yes: The software populates the CLEANSE_STATISTICS_ table with a row of statistics for each unique ENTITY_ID.No: The software does not populate the CLEANSE_STATISTICS_ table. This is the default setting. <p>The Cleanse Statistics table contains summarized data for each record with a unique value in the <code><ENTITY_ID></code> field. Use the data to gain a high-level insight into what the software changed during the cleansing process.</p> <p>Information in the table includes the total number of:</p> <ul style="list-style-type: none">RecordsSuspect recordsBlank recordsRecords with one or more significant change
	Global Address Cleanse	
Generate Cleanse Info Codes Table	Data Cleanse	<p>Specifies to populate the Cleanse Info Codes table in the repository. Options include:</p> <ul style="list-style-type: none">Yes: The software populates the CLEANSE_INFO_CODES_ table in the repository with a row for each significant information code generated by the cleanse process.No: The software does not populate the CLEANSE_INFO_CODES_ table. This is the default setting. <p>Analyze this table to find potential suspect data in your data source. Analyze the table information to view a distribution of information codes or a count of missing, suspect, or blank data.</p> <div>i Note Not all existing information codes apply to data quality statistics.</div>
	Global Address Cleanse	

Option	Transform	Description
Generate Cleanse Change Info Table	Data Cleanse	<p>Specifies to populate the Cleanse Change Info table in the repository. Options include:</p> <ul style="list-style-type: none"> • Yes: The software populates the CLEANSE_CHANGE_INFO_ table in the repository with a row for each significant change applied by the cleansing operation. • No: The software does not populate the CLEANSE_CHANGE_INFO_ table. This is the default setting. <p>Analyze this detailed table to identify concerning areas of your source data by the number of significant changes.</p> <div> ❖ Example You research a data source further when the data shows a high count of significant changes for firm or address components. </div>
	Global Address Cleanse	
Generate Cleanse Component Info Table	Data Cleanse	<p>Specifies to populate the Cleanse Component Info table. Options include:</p> <ul style="list-style-type: none"> • Yes: The software populates the CLEANSE_COMPONENT_INFO_ table in the repository. • No: The software does not populate the CLEANSE_COMPONENT_INFO_ table. This is the default setting. <p>This detailed table contains position information for each specified data element under the following circumstances:</p> <ul style="list-style-type: none"> • When it is parsed from an input field. • When it is written to an output field. • When it is parsed from an input field and written to an output field. <p>The transform can populate an output field column when there is no direct link to an input. And an input may not have any directly linked output.</p> <div> ❖ Example Two examples: <ul style="list-style-type: none"> • On input there is no prename included for a name. However, on output, the prename is included because of reference data. • On input, the transform maps to <code><given_name1></code>. And, on output, the transform does not include any <code><person1></code> output fields. </div>
	Global Address Cleanse	

Option	Transform	Description
Generate Cleanse Address Record Info Table	Global Address Cleanse	<p>Applies to Global Address Cleanse for address data. Specifies to populate the Cleanse Address Record Info table in the repository. Options include:</p> <ul style="list-style-type: none"> Yes: The software populates the CLEANSE_ADDRESS_RECORD_INFO_ table in the repository with up to two rows per record for each cleansed address entity. No: The software does not populate the CLEANSE_ADDRESS_RECORD_INFO_ table. This is the default setting. <p>Analyze this detailed table to gain information about the results of the address cleanse process on the data. The table contains statistics that focus on the results of address cleansing, which includes assignment information and assignment type for each address.</p>
Generate Geocode Statistics Table	Geocoder	<p>Specifies to populate the Geocode Statistics table in the repository. Options include:</p> <ul style="list-style-type: none"> Yes: The software populates the GEOCODE_STATISTICS_ table in the repository with one row of totals for each job execution. No: The software does not populate the GEOCODE_STATISTICS_ table. This is the default setting. <p>Analyze this table for a summary of the geocode process on the data. The table provides the total number of rows in the data and the total number of rows that were assigned a latitude and longitude.</p>
Generate Geocode Info Codes Table	Geocoder	<p>Specifies to populate the Geocode Info Codes table in the repository. Options include:</p> <ul style="list-style-type: none"> Yes: The software populates the GEOCODE_INFO_CODES_ table in the repository with one row for each significant information code assigned during the geocode process. No: The software does not populate the GEOCODE_INFO_CODES_ table. This is the default setting. <p>Use the information codes to analyze this detailed table. Track the reasons why records were not assigned the highest level of latitude and longitude.</p> <div> <p>i Note</p> <p>Not all existing information codes apply to the data quality statistics.</p> </div>

Parent topic: [Data Cleanse options \[page 476\]](#)

Related Information

[Common options \[page 477\]](#)
[Report and Analysis \[page 478\]](#)
[Cleansing Package options \[page 482\]](#)
[Options \[page 485\]](#)
[Input word breaker \[page 486\]](#)
[Person standardization options \[page 487\]](#)
[Gender standardization options \[page 491\]](#)
[Firm standardization options \[page 492\]](#)
[Other standardization options \[page 493\]](#)
[Date options \[page 496\]](#)
[Phone Options group \[page 499\]](#)
[Parser configuration \[page 501\]](#)
[Memory cache \[page 502\]](#)
[Data quality statistics common requirements \[page 449\]](#)
[Data quality statistics transform set up requirements \[page 451\]](#)
[Content information for data quality statistics tables \[page 1346\]](#)

8.5.8.4 Geocoder transform options

Set values for the Geocoder transform options to specify options such as assignment levels and threshold settings for locality and postcode assignments.

This option group is required. The options support all countries.

The following list specifies the supported directory formats based on modes:

- **Basic**: Supports the address geocoding mode. It returns centroid-level and address-level latitude and longitude information.
- **Advanced**: Supports the address geocoding, reverse geocoding, and POI textual search modes. It returns range-based and centroid-level latitude and longitude information and point-of-interest information.

The following table contains descriptions for the Geocoder transform options, and it includes the supported directory format.

Option	Description	Supported directory format
Best Assignment Level	<p>Specifies the depth of assignment for the latitude and longitude output fields. This option is used for address geocoding mode, and also for reverse geocoding with address mode.</p> <p><i>Preferred</i>: Assigns to the finest depth, which by default is to the primary number level.</p> <p><i>Primary Number</i>: Assigns to the primary number level.</p> <p><i>Postcode</i>: Assigns to the postcode level.</p> <p><i>Locality</i>: Assigns to the locality, city, or suburb level.</p> <p><i>Smallest Area</i>: The software first attempts to assign to the primary number. If there is no primary number, the software assigns based on postcode or locality information, depending on which level is the smaller area.</p> <div> <p>❖ Example</p> <p>For example, the French postcode 75014 is a smaller area than the locality of Paris.</p> </div> <div> <p>i Note</p> <p>For the <i>Preferred</i>, <i>Primary Number</i>, and <i>Smallest Area</i> assignment level values:</p> <ul style="list-style-type: none"> If the <i>Locality Assignment Threshold</i> or <i>Postcode Assignment Threshold</i> options are set to <i>None</i>, the software uses locality or postcode centroid assignment. If the <i>Locality Assignment Threshold</i> or <i>Postcode Assignment Threshold</i> options are not set to <i>None</i>, the software uses locality or postcode centroid assignment. </div>	Basic, Advanced
Distance Unit	<p>Specifies the unit of distance used for the radius. Options include:</p> <ul style="list-style-type: none"> <i>Kilometers</i> <i>Miles</i> <p>The value of the <code><Option_Distance_Unit></code> input field takes precedence over the value of the <i>Distance Unit</i> option. The transform uses the value of the <i>Distance Unit</i> option only if the <code><Option_Distance_Unit></code> input field is not mapped, is invalid, blank, or NULL.</p>	Advanced

Option	Description	Supported directory format								
Locality Assignment Threshold	<p>For address geocoding mode and reverse geocoding with address mode. Limits the level of locality centroid assignment. Specifies the locality level to use for the assignment threshold. Options include:</p> <ul style="list-style-type: none"><i>None</i>: Skips the specific assignment level. Use this setting if you do not want to return an assignment threshold on locality.<i>Preferred</i>: Assigns to the finest depth at the locality level. <p>Finest depth: Locality1 is the most general and Locality4 is the most specific.</p> <div><p>❖ Example</p><p>In this example, there is no Locality4. If you choose <i>Preferred</i>, the transform returns Locality3, which is the finest depth when there is no Locality4.</p><table><tr><th>Address</th><th>Locality level</th></tr><tr><td>Church Cottage</td><td>Locality3</td></tr><tr><td>Pemborough</td><td>Locality2</td></tr><tr><td>Bristol</td><td>Locality1</td></tr></table></div> <p>Setting the option to <i>Locality1</i> excludes <i>Locality2–4</i> during assignment, even though the software may return values at those levels. However, if you set the option to <i>Locality4</i> and there is no <i>Locality4</i> data, the transform returns the finest available data even if that data is at <i>Locality2</i> level.</p>	Address	Locality level	Church Cottage	Locality3	Pemborough	Locality2	Bristol	Locality1	Basic, Advanced
Address	Locality level									
Church Cottage	Locality3									
Pemborough	Locality2									
Bristol	Locality1									
Max Records	<p>Specifies the maximum number of records that can be returned. You can enter a number up to 100.</p> <p>The value of the <code><Option_Max_Records></code> input field takes precedence over the value of the Max Records option. The value of the Max Records option is only used if the <code><Option_Max_Records></code> input field is not mapped or is blank.</p>	Advanced								
Offset Coordinates	<p>Specifies whether the offset values of latitude and longitude are returned when the side of the street is known. This option is used for address geocoding mode, and also for reverse geocoding with address mode.</p> <p><i>Yes</i>: Returns the offset values.</p> <p><i>No</i>: Returns the center value regardless of whether the side of the street is known.</p>	Advanced								

Option	Description	Supported directory format
Postcode Assignment Threshold	<p>Limits the level of postcode centroid assignment. For example, setting the option to <i>Postcode1</i> excludes the other levels during assignment, even though the application may return values at those levels.</p> <p>This option is used for address geocoding mode, and also for reverse geocoding with address mode.</p> <p><i>Postcode Full</i>: Assigns to the entire extended postcode. For example, in the USA, it assigns to the 5-digit postcode and all four digits of the ZIP+4.</p> <p><i>Postcode1</i>: Assigns to the city or postcode area. For example, in the USA, it assigns to the 5-digit ZIP Code.</p> <p><i>Postcode2 Partial</i>: Assigns to the first few characters of the extended postcode. For example, in the USA, it assigns the 5-digit postcode and the first two digits of the ZIP+4.</p> <p><i>Preferred</i>: Assigns to the finest depth at the postcode level.</p> <p><i>None</i>: Skips the specific assignment level. Use this setting if you do not want to return an assignment threshold on postcode.</p>	Basic, Advanced
Radius	<p>The distance from a specified reference point used to identify an area in which matching records are located.</p> <p>The value of the <code><Option_Radius></code> input field takes precedence over the value of the Radius option. The value of the Radius option is only used if the <code><Option_Radius></code> input field is not mapped or is blank. If a radius is not specified, a default radius of 1 kilometer is used.</p> <p>In reverse geocoding mode, the maximum radius distance is 111 kilometers or 68.97 miles.</p>	Advanced

8.5.8.5 Reference files

Set the location for the Geocoder reference files.

Option	Description
Directory Path	<p>Specifies the path to the location of your Geocoder directory files.</p> <p>For easier set up, use the substitution variable <code>\$\$RefFiles-Geocoder</code> as a value for this option. The <code>\$\$RefFilesGeocoder</code> substitution variable is set for the default reference file location, which is the path relative to where Data Services is installed.</p> <div><p>❖ Example</p><pre><INSTALL_DIR>\Data Services \DataQuality\reference_data</pre></div> <p>If you installed Data Services to a different location, change the substitution variable while designing the work flow. For more information about variables, see the Variables and Parameters section in the <i>Data Services Designer Guide</i>.</p> <p>To set more than one location, duplicate the Directory Path option by right-clicking and selecting <i>Duplicate option</i>. Keep in mind that you cannot have the same directory file located in both locations. In the <i>Directory Path</i> option, only specify the path. Do not specify the directory file names.</p>

For information about downloading directories, see the latest Directories Update letters at https://uacp2.hana.ondemand.com/viewer/p/ADDRESSING_DIRECTORIES.

8.5.8.6 Geocoder directories

The Geocoder directories are designed specifically to use with the Geocoder transform.

SAP offers two directory formats:

- *Basic*: Supports the address geocoding mode. It returns centroid-level and address-level latitude and longitude information.
- *Advanced*: Supports the address geocoding, reverse geocoding, and POI textual search modes. It returns a range-based and centroid-level latitude and longitude information and point-of-interest information

Directory updates

A quarterly vendor-specific directory update is available on the SAP Help Portal and is also included in each directory package. The directory update provides instructions for installing the directories and information about the directory contents. It describes the package names, files names, and the supported countries, directory format, geocoding mode, centroid level, and census information. It also lists the available POI types supported by the vendor.

The Geocoder transform is flexible enough to accept new country directory data immediately after the directory data is released. There is no need to wait for the next SAP Data Services release to begin using new country directory data.

For information about downloading directories, see the latest Directories Update letters at https://uacp2.hana.ondemand.com/viewer/p/ADDRESSING_DIRECTORIES.

8.5.8.7 Input fields for the Geocoder transform

Use the Geocoder input fields to define the fields to use from your source data.

The following table contains descriptions for all of the recognized input fields that you can use in the input mapping for the Geocoder transform. It includes the applicable categories for each field. The fields are listed alphabetically.

The table also shows the input field availability based on currently supported directory formats. Supported directory formats are:

- **Basic**: Supports the address geocoding mode. It returns centroid-level and address-level latitude and longitude information.
- **Advanced**: Supports the address geocoding, reverse geocoding, and POI textual search modes. It returns range-based and centroid-level latitude and longitude information and point-of-interest information.

The input fields support all countries.

Input field name (Geocoder)	Category	Description	Supported directory format
Data_Source_ID	N/A	<p>Specifies the name of the data source that the record comes from.</p> <p>If you choose to generate data quality statistics tables, use this field as part of your mapping strategy. Also use primary key fields and the output field <Row_ID>. This mapping strategy identifies the source of individual records in any of the detailed data quality statistics tables. Use this information to trace problem data to the data source.</p> <p>Applicable for the Data Cleanse, Global Address Cleanse, and Geocoder transforms.</p>	N/A
Country	Address	The two-character ISO country code.	Basic, Advanced

Input field name (Geocoder)	Category	Description	Supported directory format
Latitude	Latitude/ Longitude	A relative distance north or south of the equator, measured in 0-90 degrees.	Advanced
Locality1-4	Address	The city, town, or suburb and any additional-related information.	Basic, Advanced
Longitude	Latitude/ Longitude	A relative distance east or west of the Greenwich meridian, measured in 0-180 degrees.	Advanced
Option_Distance_Unit	Search Filter	<p>The distance unit of measure used for the radius. Valid values are:</p> <p>Kilometers</p> <p>Miles</p> <p>The value of the <code><Option_Distance_Unit></code> input field takes precedence over the value of the Distance Unit Geocoder option. The transform uses the value of the Distance Unit option only if the <code><Option_Distance_Unit></code> input field is not mapped, is invalid, blank, or NULL.</p> <p>This field is a dynamic input field. If you change this setting, you do not have to terminate and reinitialize the transform in order for the new configuration to be recognized.</p>	Advanced
Option_Max_Records	Max Records	<p>The maximum number of records that can be returned. You can enter a number up to 100.</p> <p>A value greater than 0 outputs multiple results as XML to the <code><Result_List></code> output field rather than to individual output fields.</p> <p>The value of the <code><Option_Max_Records></code> input field takes precedence over the value of the Max Records Geocoder option. The transform uses the value of the Max Records option only if the <code><Option_Max_Records></code> input field is not mapped, is blank, or NULL.</p> <p>This field is a dynamic input field. If you change this setting, you do not have to terminate and reinitialize the transform in order for the new configuration to be recognized.</p>	Advanced

Input field name (Geocoder)	Category	Description	Supported directory format
Option_Radius	Search Filter	<p>The distance from a specified reference point used to identify an area in which matching records are located.</p> <p>In reverse geocoding mode, the maximum radius distance is 111 kilometers or 68.97 miles.</p> <p>The value of the <Option_Radius> input field takes precedence over the value of the <i>Radius</i> Geocoder option. The transform uses the value of the <i>Radius</i> option only when the <Option_Radius> input field is not mapped, is blank, or NULL.</p> <p>This field is a dynamic input field. If you change this setting, you do not have to terminate and reinitialize the transform in order for the new configuration to be recognized.</p>	Advanced
POI_Name	Address POI	The name of a point of interest, such as the Washington Monument.	Advanced
POI_Type	Address POI	<p>The point-of-interest type expressed as a number. For example, for one vendor 5999 is an historical monument.</p> <p>In POI textual search mode, to return multiple point-of-interest types, concatenate POI type codes using a colon as a delimiter. For example, to return all schools (type 8211) and libraries (type 8231) within a defined area, you would enter:</p> <p>8211 : 8231</p> <p>The POI types and their corresponding codes differ depending on the data vendor that you use. For a detailed list of available POI types, see the vendor-specific directory update.</p>	Advanced
Postcode1–2	Address	The postal code and a secondary postal code, if available.	Basic, Advanced
Primary_Name1–4	Address	The street name.	Basic, Advanced
Primary_Number	Address	The premise number.	Basic, Advanced
Primary_Postfix1	Address	Abbreviated directional (N, S, NW, SE) that follows a street name.	Basic, Advanced
Primary_Prefix1	Address	Abbreviated directional (N, S, NW, SE) that precedes a street name.	Basic, Advanced
Primary_Type1–4	Address	Abbreviated type of primary name (St., Ave., or Pl.).	Basic, Advanced
Region1–2	Address	The region symbol of the state, province, or territory.	Basic, Advanced
Search_Filter_Name	Search Filter	Search criteria for a point-of-interest name.	Advanced

Input field name (Geocoder)	Category	Description	Supported directory format
Search_Filter_Type	Search Filter	<p>Search criteria for a point-of-interest type, expressed as a four-digit number; for example, for one vendor 5999 is historical monument.</p> <p>If you want to return an address only, enter ADDR.</p> <p>To return multiple point-of-interest types, concatenate POI type codes using a colon as a delimiter. For example, to return all schools (type 8211) and libraries (type 8231) within a defined area, you would enter:</p> <p>8211 : 8231</p> <p>To return a point-of-interest type and its address, enter:</p> <p>5999 : ADDR</p> <p>The POI types and their corresponding codes differ depending on the data vendor that you use. For a detailed list of available POI types, see the vendor-specific directory update.</p>	Advanced

Related Information

[Dynamic transform settings \[page 354\]](#)

8.5.8.8 Output fields for the Geocoder transform

The following are recognized output fields that you can use in the output mapping for the Geocoder transform. The fields are listed alphabetically. The table also shows the output field availability based on currently supported countries and directory formats.

Supported countries:

- [All](#): All countries
- [USA](#)

Supported directory formats:

- [Basic](#): Supports the address geocoding mode. It returns centroid-level and address-level latitude and longitude information.
- [Advanced](#): Supports the address geocoding, reverse geocoding, and POI textual search modes. It returns range-based and centroid-level latitude and longitude information and point-of-interest information.

Output field name (Geocoder)	Category	Description	Supported countries	Supported directory format
Address_Line	Address	A line of data in an address that contains the primary address. The primary address can contain components such as the primary range, primary name, directionals (post- and pre-), and suffix.	All	Basic, Advanced
Assignment_Level	Assignment Level	<p>The level to which the transform matches the address to the data in the reference fields (directories).</p> <p><i>PRE</i>: Primary Range Exact assigns to the exact location of the address; for example, 123 Main St. This is the most precise level of assignment. To obtain the PRE, you must map either the POI_Type input field or the Primary_Name and Primary_Number input fields.</p> <p><i>PR</i>: Primary Range Interpolated assigns to the level of the address range; for example, 100-500 Main St.</p> <p><i>L1-4</i>: Locality1-4 assigns to the level of city, town, or suburb.</p> <p><i>P1</i>: Postcode1 assigns to the level of Postcode1.</p> <p><i>P2P</i>: Postcode2 Partial assigns the full Postcode1 and the first few characters of Postcode2.</p> <p><i>PF</i>: Postcode Full assigns to the level of Postcode1 and Postcode2, when available.</p>	All	Basic, Advanced
Assignment_Level_Locality	Assignment Level	<p>The level to which the transform assigns the locality.</p> <p><i>L1-4</i>: Returns up to four locality levels. L1 is the most general and L4 is the most specific.</p>	All	Basic, Advanced

Output field name (Geocoder)	Category	Description	Supported countries	Supported directory format
Assignment_Level_Postcode	Assignment Level	<p>The level to which the transform assigns the postcode.</p> <p><i>P1</i>: Postcode1 assigns to the level of Postcode1.</p> <p><i>P2P</i>: Postcode2 Partial assigns the full Postcode1 and the first few characters of Postcode2.</p> <p><i>PF</i>: Postcode Full assigns to the level of Postcode1 and Postcode2, when available.</p>	All	Basic, Advanced
Census_Tract_Block	Census	The census tract code as defined by the government for reporting census information. Census tracts are small, relatively permanent statistical subdivisions of a county.	USA	Basic, Advanced
Census_Tract_Block_Prev	Census	The census tract code in the previous version of census data.	N/A	N/A
Census_Tract_Block_Group	Census	The census tract block group code as defined by the government for reporting census information. These codes are used for matching with demographic-coding databases. In the USA, the first six digits contain the tract number (for example, 002689); the next digit contains the block group (BG) number within the tract, and the last three digits contain the block code. The BG is a cluster of census blocks that have the same first digit within a census tract. For example, BG 6 includes all blocks numbered from 6000 to 6999.	USA	Basic, Advanced
Census_Tract_Block_Group_Prev	Census	The census tract block group code in the previous version of census data.	N/A	N/A
Country_Code	Address	The two-character ISO country code.	All	Basic, Advanced
Distance	Distance	The distance from the input address, geographical coordinates, or point of interest to the closest address or point of interest.	All	Advanced
Gov_County_Code	Census	A unique county code as defined by the government for reporting census information. For example, in the USA, this is a Federal Information Processing Standard (FIPS) three-digit county code.	USA	Basic, Advanced

Output field name (Geocoder)	Category	Description	Supported countries	Supported directory format
Gov_Locality1_Code	Census	A unique code for an incorporated municipality such as a city, town, or locality as defined by the government for reporting census information.	USA	Basic
Gov_Region1_Code	Census	A unique region code as defined by the government for reporting census information. For example, in the USA, this is a Federal Information Processing Standard (FIPS) two-digit state code.	USA	Basic, Advanced
Info_Code	Info Code	A three-character code that provides information about the geocoding results. The status for address and point-of-interest geocoding assignment is indicated in the third character. The status for reverse geocoding assignment is indicated in the second and third characters. If assigned to the best level, the Info_Code field is blank. The first character is not used at this time. For more information, see Information codes [page 548] .	All	Basic, Advanced
Latitude	Latitude/Longitude	The latitude at the best assigned level (0–90 degrees north or south of the equator) standardized to six decimals in the format 45.801357.	All	Basic, Advanced
Latitude_Locality	Latitude/Longitude	The latitude at the locality level centroid of the city, town, locality, or suburb standardized to six decimals in the format 45.801357.	All	Basic, Advanced
Latitude_Postcode	Latitude/Longitude	The latitude at the postcode level centroid of the postcode standardized to six decimals in the format 45.801357.	All	Basic, Advanced
Latitude_Primary_Number	Latitude/Longitude	The latitude at the primary number level standardized to six decimals in the format 45.801357.	All	Basic, Advanced
Locality1–4	Address	The city, town, or suburb and any additional related information.	All	Basic, Advanced
Longitude	Latitude/Longitude	The longitude at the best assigned level (0–180 degrees east or west of Greenwich meridian) standardized to six decimals in the format 123.458331.	All	Basic, Advanced
Longitude_Locality	Latitude/Longitude	The longitude at the locality level centroid of the city, town, locality, or suburb standardized to six decimals in the format 123.458331.	All	Basic, Advanced

Output field name (Geocoder)	Category	Description	Supported countries	Supported directory format
Longitude_Postcode	Latitude/Longitude	The longitude at the postcode level centroid of the postcode standardized to six decimals in the format 123.45833.	All	Basic, Advanced
Longitude_Primary_Number	Latitude/Longitude	The longitude at the primary number level standardized to six decimals in the format 123.458331.	All	Basic, Advanced
Metro_Stat_Area_Code	Census	The metropolitan statistical area. For example, in the USA, the 0000 code indicates the address does not lie in a metropolitan statistical area; usually a rural area. A metropolitan statistical area has a large population that has a high degree of social and economic integration with the core of the area. The area is defined by the government for reporting census information.	USA	Basic
Metro_Stat_Area_Code_Prev	Census	The metropolitan statistical area in the previous version of census data.	N/A	N/A
Minor_Div_Code	Census	The minor civil division or census county division code when the minor civil division is not available. The minor civil division designates the primary government and/or administrative divisions of a county such as a civil township or precinct. Census county division are defined in a state or province that does not have a well-defined minor civil division. The area is defined by the government for reporting census information.	USA	Basic, Advanced
Minor_Div_Code_Prev	Census	The minor civil division or census county division code in the previous version of census data.	N/A	N/A
POI_Name	Address POI	The point of interest name, such as the Washington Monument.	All	Advanced
POI_Type	Address POI	The point of interest type expressed as a four-digit number; for example, 5999 (historical monument).	All	Advanced

Output field name (Geocoder)	Category	Description	Supported countries	Supported directory format
Population_Class_Locality1	Population	<p>Indicates that the population falls within a certain size.</p> <p>0: Undefined. The population may be too large or small to provide accurate data.</p> <p>1: Over 1 million.</p> <p>2: 500,000 to 999,999.</p> <p>3: 100,000 to 499,999.</p> <p>4: 50,000 to 99,999.</p> <p>5: 10,000 to 49,999.</p> <p>6: Less than 10,000.</p>	All	Advanced
Postcode	Address	The postal code.	All	Basic, Advanced
Postcode1-2	Address	The postal code and a secondary postal code, if available.	All	Basic, Advanced
Primary_Name1-4	Address	The street name.	All	Basic, Advanced
Primary_Number	Address	The premise number.	All	Basic, Advanced
Primary_Postfix1	Address	Abbreviated directional (N, S, NW, SE) that follows a street name.	All	Basic, Advanced
Primary_Prefix1	Address	Abbreviated directional (N, S, NW, SE) that precedes a street name.	All	Basic, Advanced
Primary_Range_High	Address	The high value of a primary number range.	All	Basic, Advanced
Primary_Range_Low	Address	The low value of a primary number range.	All	Basic, Advanced
Primary_Type1-4	Address	Abbreviated type of primary name (St., Ave., or Pl.).	All	Basic, Advanced
Region1-2	Address	The region symbol of the state, province, or territory.	All	Basic, Advanced
Result_List	Results	The XML output when multiple records are returned for a search.	All	Advanced
Result_List_Count	Results	The number of results in the Result_List output field.	All	Advanced

Output field name (Geocoder)	Category	Description	Supported countries	Supported directory format
Row_ID	Statistics	<p>An internal ID generated by the transform that uniquely identifies a row processed by that transform.</p> <p>Use this field to join the input and output to the applicable non-summary data quality statistics table.</p> <p>The software issues a warning if you select to generate any data quality statistics tables in the transform, and you don't include the Row_ID output field.</p> <p>The software also issues a warning if you map this output field and don't select to generate any non-summary data cleanse statistics tables in the transform.</p> <p>Applicable transforms are Data Cleanse, Global Address Cleanse, and Geocoder.</p>	N/A	N/A
Side_Of_Primary_Address	Side of Street	Indicates that the location is on the L (left) or R (right) side of the street when moving north, northeast, northwest, or east.	All	Advanced
Stat_Area_Code	Census	A core-based statistical area code where an area has a high degree of social and economic integration within the core that the area surrounds. The area is defined by the government for reporting census information.	USA	Basic
Stat_Area_Code_Prev	Census	The statistical area code in the previous version of census data.	N/A	N/A
Status_Code	Status Code	Displays a four-character status code that always starts with an S. This code represents the fuzzy matches made to the address/point of interest (POI) component during address geocoding processing. For more information, see Status codes [page 550] .	All	Advanced, Basic

Related Information

[Information codes \[page 548\]](#)

[Status codes \[page 550\]](#)

8.5.8.9 Result_List XML output fields

Map `<Result_List>` XML output fields to include the information in your output data.

The following are recognized output fields that you can use in the `<Result_List>` XML output field in the Geocoder transform. The fields are listed alphabetically.

Output field name	Category	Description
Address_Line	Address	A line of data in an address that contains the primary address. The primary address can contain components such as the primary range, primary name, directionals (post- and pre), and suffix.
Assignment_Level	Assignment Level	<p>The level to which this transform matched the address to the data in the reference fields (directories).</p> <p><i>PRE</i>: Primary Range Exact assigns to the exact location of the address; for example, 123 Main St. Primary Range Exact is the most precise level of assignment. To obtain the PRE, map either the <code><POI_Type></code> input field or the <code><Primary_Name></code> and <code><Primary_Number></code> input fields.</p> <p><i>PRI</i>: Primary Range Interpolated assigns to the level of the address range; for example, 100-500 Main St.</p> <p><i>L1-4</i>: Locality1-4 assigns to the level of city, town, or suburb.</p> <p><i>P1</i>: Postcode1 assigns to the level of Postcode1.</p> <p><i>P2P</i>: Postcode2 Partial assigns the full Postcode1 and the first few characters of Postcode2.</p> <p><i>PF</i>: Postcode Full assigns to the level of Postcode1 and Postcode2, when available.</p>
Country_Code	Address	The two-character ISO country code.
Distance	Distance	The distance from the input address, geographical coordinates, or point of interest to the closest address or point of interest.
Group_Number	Group	An identifier for multiple records that represent the same address. For example, one record might have an alias primary name of another. This output field has a value only when the same address is returned in multiple entries. When entries have the same value in Group_Number, it means that the entries have the same address.

Output field name	Category	Description
Group_Rank	Group	<p>Indicates whether a record in a group is a master or a subordinate. For each group of entries that have the same Group_Number value, one of them has Group_Rank of "M" (master) and all other entries in the group have "S" (subordinate). You may obtain only one of each address returned by filtering where the Group_Rank is either "M" or a blank string.</p> <p><i>M</i>: Master</p> <p><i>S</i>: Subordinate</p>
Latitude	Latitude/Longitude	The latitude at the best assigned level (0–90 degrees north or south of the equator) in the format 45.32861.
Locality1–4	Address	The city, town, or suburb and any additional related information.
Longitude	Latitude/Longitude	The longitude at the best assigned level (0–180 degrees east or west of Greenwich meridian) in the format 123.45833.
POI_Name	Address POI	The point of interest name, such as the Washington Monument.
POI_Type	Address POI	The point of interest type expressed as a four-digit number; for example, 5999 (historical monument).
Postcode	Address	The postal code.
Postcode1–2	Address	The postal code and a secondary postal code, if available.
Primary_Name1–4	Address	The street name.
Primary_Number	Address	The premise number.
Primary_Postfix1	Address	Abbreviated directional (N, S, NW, SE) that follows a street name.
Primary_Prefix1	Address	Abbreviated directional (N, S, NW, SE) that precedes a street name.
Primary_Range_High	Address	The high value of a primary number range.
Primary_Range_Low	Address	The low value of a primary number range.
Primary_Type1–4	Address	Abbreviated type of primary name (St., Ave., or Pl.).
Ranking	Ranking	A numeric value that indicates how well the returned records match the input field based on the match score. A record with a ranking of 1 has the highest match score.
Region1–2	Address	The region symbol of the state, province, or territory.

❖ Example

The following is an example for an XML result list that has one record:

```
<RESULT_LIST>
  <RECORD>
    <ADDRESS_LINE>332 FRONT ST</ADDRESS_LINE>
    <ASSIGNMENT_LEVEL>PRE</ASSIGNMENT_LEVEL>
    <COUNTRY_CODE>US</COUNTRY_CODE>
    <DISTANCE>0.3340</DISTANCE>
```

```

<LATITUDE>43.811616</LATITUDE>
<LOCALITY1>LA CROSSE</LOCALITY1>
<LONGITUDE>-91.256695</LONGITUDE>
<POI_NAME>ABC COMPANY</POI_NAME>
<POI_TYPE>5800</POI_TYPE>
<POSTCODE>56001-4023</POSTCODE>
<POSTCODE1>56001</POSTCODE1>
<POSTCODE2>4023</POSTCODE2>
<PRIMARY_NAME1>FRONT</PRIMARY_NAME1>
<PRIMARY_NUMBER>332</PRIMARY_NUMBER>
<PRIMARY_TYPE1>ST</PRIMARY_TYPE1>
<RANKING>1</RANKING>
<REGION1>WI</REGION1>
</RECORD>
</RESULT_LIST>

```

8.5.8.10 Information codes

The transform assigns information codes to a record when you include the `<Info_Code>` output field in your field mapping.

The Info_Code output field is a three-character code that provides information about geocoding results.

The third character of the information code indicates the status for the address geocoding and POI textual search modes. The second and third characters of the information code indicates the status for reverse geocoding. If you set the assignment to the best level, the `<Info_Code>` field is blank. The first character is not used at this time.

Use the following table to understand the code assigned to the `<Info_Code>` output field.

Information code	Description
001	Reference data is not available for the input country. Verify that the directory is installed and the reference path to the directory is valid.
002	Address-level reference data is not available for the input data. When Best Assignment Level is set to Primary Number and the address directory is unavailable or doesn't exist, this code is output. Verify that the directory is installed and the reference path to the directory is valid.
003	Centroid-level reference data is not available for the input data. When the Best Assignment Level is set to Locality or Postcode, and the address directory is unavailable or doesn't exist, this code is output. Verify that the directory is installed and the reference path to the directory is valid.
004	Assignment is limited. The input data is insufficient or incorrect to match the reference data. When the Best Assignment Level fails, this code is output. The assignment may be made to a lower assignment level than the one specified. For example, if you set Best Assignment Level to Primary Number and the Primary Number field is blank, the assignment may be at the postcode or locality level, if the data is available. Verify your input data and input field mapping and make sure that the fields required for best assignment exist and are correctly mapped.

Information code	Description
005	<p>The input data does not match anything in the reference data. When the input record does not match the directory data for the Best Assignment Level or a lower assignment level, this code is output.</p> <p>Verify your input data and input field mapping and make sure that the fields required for best assignment exist and are correctly mapped.</p>
006	<p>The input data assigns ambiguously in the reference data. There is a tie for the Best Assignment Level. The input record matches several records in the directory data and the software cannot decide which one is the best.</p> <p>For example, if the reference data has two records:</p> <p>Record 1: 100 Main St La Crosse WI 54650</p> <p>Record 2: 100 Main St Bt Micts WI 54650</p> <p>When input with 100 Main ST WI 54650 without a locality name, the 006 information code for ambiguous assignment is output.</p> <p>Verify your input data and input field mapping and make sure that the fields required for best assignment exist and are correctly mapped.</p>
007	<p>The input data is blank or invalid.</p> <p>For example, if the US Postcode1 is a five-digit string and your input data is a six-digit string, the 007 information code is output.</p> <p>Verify your input data and input field mapping and make sure that the fields required for best assignment exist and are correctly mapped.</p>
008	<p>The input data is insufficient. When the input data for the selected Best Assignment Level is blank, this code is output. For example, this code is output when you set the Best Assignment Level to Primary Number and the input data is blank for Primary Number.</p>
009	<p>The POI type provided on input is invalid. The point-of-interest type is not correct.</p> <p>Verify your input POI type with the POI types described in the appropriate directory update letter.</p>
050	<p>Reverse geocoding. The input data does not match anything in the reference data.</p>
060	<p>Reverse geocoding. The input data assigns ambiguously in the reference data.</p>
070	<p>Reverse geocoding. The input data is blank or invalid.</p>
090	<p>Reverse geocoding. The POI type provided on input is invalid.</p>
00A	<p>The POI input data was not used. The POI name or type does not match the directory data. A PRI or PRE level assignment was made; however, the input POI name and POI type were not used for the assignment.</p>
0B0	<p>The input data was not found. This code only occurs when an address is input during reverse geocoding assignment. The input address doesn't match the directory data and reverse geocoding cannot be performed based on the address.</p>

Information code	Description
OCO	<p>Not all results returned for the input data, because the number of results exceeds the specified Max_Record. This code only occurs during reverse geocoding assignment.</p> <p>Increase the Max_Record value.</p>
ODO	<p>Not all results were returned for the input data, because the results exceed the field length available in the Result_List XML output field. This code only occurs during reverse geocoding assignment. The Geocoder transform allows a 60000 field length in the Result_List field.</p>
OOE	<p>The closest latitude and longitude to the input address is returned because the input house number does not exist in the geocoder directories.</p> <p>For example:</p> <p>Input address: 100 Main St La Crosse WI 54650</p> <p>Directory data: 1-88 Main St La Crosse WI 54650</p> <p>The software returns the latitude and longitude values for 88 Main St with the OOE information code to indicate that the house number does not match the directory data.</p>
OFO	<p>The feature enabled by the current configuration is not supported by the installed basic directory. For example, a configuration may request reverse geocoding but it only finds the basic directory installed.</p> <p>Verify that the advanced directory format is installed and the reference path to the directory is valid. For more information about directories, see "Directory data" in the <i>SAP Data Services Installation Guide</i>.</p>

8.5.8.11 Status codes

The transform assigns status codes to a record when you include the `<status_Code>` output field in your field mapping.

Status codes are four characters that represent the fuzzy matches made to the address—point of interest (POI) component during address geocoding processing. Each character represents an aspect of the status:

- The first character is always S (for Status).
- The second character is associated with any last-line fuzzy matches to the postal code, region, or locality.
- The third character is associated with any address-line fuzzy matches to the primary name prefix, primary name postfix, primary type, primary name, and primary range.
- The fourth character is associated with any POI data fuzzy matches.

Note

The software uses the `<Status_Code>` output field only for the address geocoding process.

Second character status code descriptions

Value	Description
0	No significant difference between the input data and the reference data.

Value	Description
1	Fuzzy match on postal code.
2	Fuzzy match on region.
3	Fuzzy match on postal code and region.
4	Fuzzy match on locality.
5	Fuzzy match on postal code and locality.
6	Fuzzy match on region and locality.
7	Fuzzy match on postal code, region, and locality.

Third character status code descriptions

Value	Description
0	No significant difference between the input data and the reference data.
1	Fuzzy match on primary name prefix/primary name postfix.
2	Fuzzy match on primary type.
3	Fuzzy match on primary name prefix/primary name postfix and primary type.
4	Fuzzy match on primary name.
5	Fuzzy match on primary name prefix/primary name postfix and primary name.
6	Fuzzy match on primary type and primary name.
7	Fuzzy match on primary name prefix/primary name postfix, primary type, and primary name.
8	Fuzzy match on primary range.
9	Fuzzy match on primary name prefix/primary name postfix and primary range.
A	Fuzzy match on primary type and primary range.
B	Fuzzy match on primary name prefix/primary name postfix, primary type, and primary range.
C	Fuzzy match on primary name and primary range.
D	Fuzzy match on primary name prefix/primary name postfix, primary name, and primary range.
E	Fuzzy match on primary type, primary name, and primary range.
F	Fuzzy match on primary name prefix/primary name postfix, primary type, primary name, and primary range.


Fourth character status code descriptions

Value	Description
0	No significant difference between the input data and the reference data.
1	Fuzzy match on POI name.

8.5.9 Global Address Cleanse

Use the Global Address Cleanse transform to identify, parse, validate, and correct your global address data.

Global Address Cleanse information

Characteristic	Description
	Global Address Cleanse icon
Use	<p>The Global Address Cleanse transform identifies, parses, validates, and corrects global address data, such as primary number, primary name, primary type, directional, secondary identifier, secondary number, locality, region, and postcode.</p> <p>The transform has options to enable generation of data quality statistics tables in the repository. These statistics provide insight into how the software cleansed and assigned your data by generating aggregated summary statistics and specific record-level statistics.</p> <div>Note The Global Address Cleanse transform does not support CASS certification or produce a USPS Form 3553. If you want to certify your U.S. address data, use the USA Regulatory Address Cleanse transform, which supports CASS.</div> <p>If you perform both address cleansing and data cleansing, the Global Address Cleanse transform typically comes before the Data Cleanse transform in the data flow.</p>
Content objects	The Global Address Cleanse transform has sample transform configurations that will help you to set up your data flow. The transforms include all of the required options except input fields.

[Report and Analysis \[page 554\]](#)

Use this option to generate report data for the Global Address Cleanse transform.

[Reference files \[page 558\]](#)

Set the location for the global reference files.

[Country ID options \(Global Address Cleanse\) \[page 559\]](#)

Set Country ID options to specify how the transform uses country ID processes.

[Engines \[page 561\]](#)

The Global Address Cleanse transform refers to country-specific processes as the engine it uses to process your data.

[Standardization Options: Country \[page 562\]](#)

Select a country name to determine the address standards the Global Address Cleanse transform uses.

[Standardization options \[page 563\]](#)

Set Standardization options to standardize address data based on the country that you select, or for all countries.

[Canada options: Global Address Cleanse \[page 572\]](#)

Set processing options in the Canada group of options for cleansing addresses from Canada.

[Canada options: Report options \[page 575\]](#)

Set the report options so that the software produces the Statement of Address Accuracy report information for SERP certification.

[Canada engine Suggestion List options \[page 576\]](#)

[Global Address \[page 578\]](#)

The Global Address group includes options for specified countries, or for global addresses.

[USA: Global Address Cleanse \[page 589\]](#)

Use the USA options in the Global Address Cleanse transform to set processing options for the United States of America and territories.

[Suggestion List: Global Address Cleanse \[page 592\]](#)

Select the components and the format for the transform suggestion list string.

[Global Address Cleanse fields \[page 597\]](#)

[Global Address Cleanse sample configurations \[page 638\]](#)

Parent topic: [Data Quality transforms \[page 437\]](#)

Related Information

[Blueprints and other content objects for download \[page 438\]](#)

[About Data Quality fields \[page 445\]](#)

[About data quality statistics \[page 448\]](#)

[Associate \[page 454\]](#)

[Country ID \[page 472\]](#)

[Data Cleanse \[page 474\]](#)

[DSF2® Walk Sequencer \[page 514\]](#)

[Geocoder \[page 524\]](#)

[Global Suggestion List \[page 640\]](#)
[Match \[page 651\]](#)
[USA Regulatory Address Cleanse \[page 710\]](#)
[Address Cleanse reference \[page 765\]](#)
[Data Cleanse reference \[page 823\]](#)
[Blueprints and other content objects for download \[page 438\]](#)
[Transform configurations \[page 440\]](#)
[Address Cleanse reference \[page 765\]](#)
[Data Quality Statistics Settings \[page 479\]](#)

8.5.9.1 Report and Analysis

Use this option to generate report data for the Global Address Cleanse transform.

Option	Description
Generate Report Data	<p>Specifies whether to generate report data for this transform.</p> <p><i>Yes</i>: Generates report data for this transform.</p> <p><i>No</i>: Turns off report data generation. If you do not need to generate reports (during testing, for example), set this option to <i>No</i> to improve performance.</p>

Parent topic: [Global Address Cleanse \[page 552\]](#)

Related Information

[Reference files \[page 558\]](#)
[Country ID options \(Global Address Cleanse\) \[page 559\]](#)
[Engines \[page 561\]](#)
[Standardization Options: Country \[page 562\]](#)
[Standardization options \[page 563\]](#)
[Canada options: Global Address Cleanse \[page 572\]](#)
[Canada options: Report options \[page 575\]](#)
[Canada engine Suggestion List options \[page 576\]](#)
[Global Address \[page 578\]](#)
[USA: Global Address Cleanse \[page 589\]](#)
[Suggestion List: Global Address Cleanse \[page 592\]](#)
[Global Address Cleanse fields \[page 597\]](#)
[Global Address Cleanse sample configurations \[page 638\]](#)

8.5.9.1.1 Data Quality Statistics Settings

The Data Quality Statistics Settings control the generation of data quality statistics tables.

The *Data Quality Statistics Settings* options in the following table apply to either Data Cleanse, Global Address Cleanse, or both transforms. Some also apply to the Geocoder transform. The following table includes a column that indicates the applicable transforms.

Data quality statistics option descriptions

Option	Transform	Description
<i>Generate Cleanse Statistics Table</i>	Data Cleanse	Specifies to generate the Cleanse Statistics table. Options include:
	Global Address Cleanse	<ul style="list-style-type: none">Yes: The software populates the CLEANSE_STATISTICS_ table with a row of statistics for each unique ENTITY_ID.No: The software does not populate the CLEANSE_STATISTICS_ table. This is the default setting. <p>The Cleanse Statistics table contains summarized data for each record with a unique value in the <ENTITY_ID> field. Use the data to gain a high-level insight into what the software changed during the cleansing process.</p> <p>Information in the table includes the total number of:</p> <ul style="list-style-type: none">RecordsSuspect recordsBlank recordsRecords with one or more significant change
<i>Generate Cleanse Info Codes Table</i>	Data Cleanse	Specifies to populate the Cleanse Info Codes table in the repository. Options include:
	Global Address Cleanse	<ul style="list-style-type: none">Yes: The software populates the CLEANSE_INFO_CODES_ table in the repository with a row for each significant information code generated by the cleanse process.No: The software does not populate the CLEANSE_INFO_CODES_ table. This is the default setting. <p>Analyze this table to find potential suspect data in your data source. Analyze the table information to view a distribution of information codes or a count of missing, suspect, or blank data.</p> <div>i Note Not all existing information codes apply to data quality statistics.</div>

Option	Transform	Description
Generate Cleanse Change Info Table	Data Cleanse	<p>Specifies to populate the Cleanse Change Info table in the repository. Options include:</p> <ul style="list-style-type: none"> • Yes: The software populates the CLEANSE_CHANGE_INFO_ table in the repository with a row for each significant change applied by the cleansing operation. • No: The software does not populate the CLEANSE_CHANGE_INFO_ table. This is the default setting. <p>Analyze this detailed table to identify concerning areas of your source data by the number of significant changes.</p> <div> ❖ Example You research a data source further when the data shows a high count of significant changes for firm or address components. </div>
	Global Address Cleanse	
Generate Cleanse Component Info Table	Data Cleanse	<p>Specifies to populate the Cleanse Component Info table. Options include:</p> <ul style="list-style-type: none"> • Yes: The software populates the CLEANSE_COMPONENT_INFO_ table in the repository. • No: The software does not populate the CLEANSE_COMPONENT_INFO_ table. This is the default setting. <p>This detailed table contains position information for each specified data element under the following circumstances:</p> <ul style="list-style-type: none"> • When it is parsed from an input field. • When it is written to an output field. • When it is parsed from an input field and written to an output field. <p>The transform can populate an output field column when there is no direct link to an input. And an input may not have any directly linked output.</p> <div> ❖ Example Two examples: <ul style="list-style-type: none"> • On input there is no prename included for a name. However, on output, the prename is included because of reference data. • On input, the transform maps to <code><given_name1></code>. And, on output, the transform does not include any <code><person1></code> output fields. </div>
	Global Address Cleanse	

Option	Transform	Description
Generate Cleanse Address Record Info Table	Global Address Cleanse	<p>Applies to Global Address Cleanse for address data. Specifies to populate the Cleanse Address Record Info table in the repository. Options include:</p> <ul style="list-style-type: none"> Yes: The software populates the CLEANSE_ADDRESS_RECORD_INFO_ table in the repository with up to two rows per record for each cleansed address entity. No: The software does not populate the CLEANSE_ADDRESS_RECORD_INFO_ table. This is the default setting. <p>Analyze this detailed table to gain information about the results of the address cleanse process on the data. The table contains statistics that focus on the results of address cleansing, which includes assignment information and assignment type for each address.</p>
Generate Geocode Statistics Table	Geocoder	<p>Specifies to populate the Geocode Statistics table in the repository. Options include:</p> <ul style="list-style-type: none"> Yes: The software populates the GEOCODE_STATISTICS_ table in the repository with one row of totals for each job execution. No: The software does not populate the GEOCODE_STATISTICS_ table. This is the default setting. <p>Analyze this table for a summary of the geocode process on the data. The table provides the total number of rows in the data and the total number of rows that were assigned a latitude and longitude.</p>
Generate Geocode Info Codes Table	Geocoder	<p>Specifies to populate the Geocode Info Codes table in the repository. Options include:</p> <ul style="list-style-type: none"> Yes: The software populates the GEOCODE_INFO_CODES_ table in the repository with one row for each significant information code assigned during the geocode process. No: The software does not populate the GEOCODE_INFO_CODES_ table. This is the default setting. <p>Use the information codes to analyze this detailed table. Track the reasons why records were not assigned the highest level of latitude and longitude.</p> <div> <p>i Note</p> <p>Not all existing information codes apply to the data quality statistics.</p> </div>

Parent topic: [Data Cleanse options \[page 476\]](#)

Related Information

[Common options \[page 477\]](#)
[Report and Analysis \[page 478\]](#)
[Cleansing Package options \[page 482\]](#)
[Options \[page 485\]](#)
[Input word breaker \[page 486\]](#)
[Person standardization options \[page 487\]](#)
[Gender standardization options \[page 491\]](#)
[Firm standardization options \[page 492\]](#)
[Other standardization options \[page 493\]](#)
[Date options \[page 496\]](#)
[Phone Options group \[page 499\]](#)
[Parser configuration \[page 501\]](#)
[Memory cache \[page 502\]](#)
[Data quality statistics common requirements \[page 449\]](#)
[Data quality statistics transform set up requirements \[page 451\]](#)
[Content information for data quality statistics tables \[page 1346\]](#)

8.5.9.2 Reference files

Set the location for the global reference files.

Reference files are directories required by the Global Address Cleanse transform to process your data.

Option	Description
Directory Path	<p>Specifies the path to the location of your Global Address directory files.</p> <p>For easier setup, use the substitution variable <code>\$\$RefFilesAddressCleanse</code> as a value for this option. The <code>\$\$RefFilesAddressCleanse</code> substitution variable is set for the default reference file location, which is the path relative to where Data Services is installed. If you installed to a location that wasn't the default location, you can change the substitution variable dynamically.</p>

For information about downloading directories, see the latest Directories Update letters at https://uacp2.hana.ondemand.com/viewer/p/ADDRESSING_DIRECTORIES.

Parent topic: [Global Address Cleanse \[page 552\]](#)

Related Information

[Report and Analysis \[page 554\]](#)
[Country ID options \(Global Address Cleanse\) \[page 559\]](#)
[Engines \[page 561\]](#)
[Standardization Options: Country \[page 562\]](#)
[Standardization options \[page 563\]](#)
[Canada options: Global Address Cleanse \[page 572\]](#)
[Canada options: Report options \[page 575\]](#)
[Canada engine Suggestion List options \[page 576\]](#)
[Global Address \[page 578\]](#)
[USA: Global Address Cleanse \[page 589\]](#)
[Suggestion List: Global Address Cleanse \[page 592\]](#)
[Global Address Cleanse fields \[page 597\]](#)
[Global Address Cleanse sample configurations \[page 638\]](#)

8.5.9.3 Country ID options (Global Address Cleanse)

Set Country ID options to specify how the transform uses country ID processes.

This option group is required.

Option	Description
Country ID Mode	<p>Specifies whether to always use the specified Country Name or to run Country ID processing.</p> <p><i>Constant:</i> Assumes all of your input data is for the specified Country Name and does not run Country ID processing. Choose this option only if all of your data is from one country, such as Australia. This option may save processing time.</p> <p><i>Assigned:</i> Runs Country ID processing. Choose this option if the input data contains addresses from more than one country.</p>

Option	Description
Country Name	<p>Specifies the country of destination.</p> <p>Select the applicable country from the dropdown list.</p> <p>Select <i>None</i> under the following circumstances:</p> <ul style="list-style-type: none"> When you set the <i>Country ID Mode</i> option to <i>Assigned</i>. When you do not want the transform to populate this option with the default country. The transform assigns the default country when it cannot identify the input country. <p>Special considerations:</p> <ul style="list-style-type: none"> If you set the <i>Country ID Mode</i> to <i>Constant</i>, choose the country of destination from the <i>Country Name</i> list. The transform assumes that all of your data is for this country. <div> <p>i Note</p> <p>You cannot choose <i>None</i> if the <i>Country ID Mode</i> option is set to <i>Constant</i>.</p> </div> <ul style="list-style-type: none"> If you set the <i>Country ID Mode</i> to <i>Assigned</i>, choose a country name to be used when the <i>Country ID</i> could not identify a country. If you set the <i>Country Name</i> to <i>None</i>, then the transform sends the address to the default engine, Global Address, for processing.
Script Code	<p>Specifies the ISO four-character script code for your data. Select one of the following options:</p> <ul style="list-style-type: none"> <i>CJKK</i>: Chinese, Japanese, and Korean <i>GREK</i>: Greek <i>LATN</i>: Latin <i>None</i>: If this option is selected, the transform attempts to identify the overall script of the input data as CJKK, GREK, LATN, or other.

Parent topic: [Global Address Cleanse \[page 552\]](#)

Related Information

[Report and Analysis \[page 554\]](#)

[Reference files \[page 558\]](#)

[Engines \[page 561\]](#)

[Standardization Options: Country \[page 562\]](#)

[Standardization options \[page 563\]](#)

[Canada options: Global Address Cleanse \[page 572\]](#)

[Canada options: Report options \[page 575\]](#)

[Canada engine Suggestion List options \[page 576\]](#)

[Global Address \[page 578\]](#)

[USA: Global Address Cleanse \[page 589\]](#)

[Suggestion List: Global Address Cleanse \[page 592\]](#)

[Global Address Cleanse fields \[page 597\]](#)

8.5.9.4 Engines

The Global Address Cleanse transform refers to country-specific processes as the engine it uses to process your data.

Before you set options in this group, consider the countries represented in your data. For example, if your data does not contain addresses from Canada or USA, select the Global Address engine.

The Global Address Cleanse transform requires that you select one or more of the Global Address Cleanse engines to process your data.

This option group is required.

Option	Description
Canada	<p>Specifies whether the transform uses the Canada engine for the Global Address Cleanse transform.</p> <p><i>Yes:</i> Activates the Canada engine.</p> <p><i>No:</i> Disables the Canada engine.</p>
Dynamic Engine Init	<p>Specifies whether engines that the transform uses are limited to the engines that it can initialize.</p> <p><i>Yes:</i> Uses only the engines that it can successfully initialize. If an engine fails to initialize, the software issues a warning and the job continues.</p> <p><i>No:</i> Uses all enabled engines. If an engine fails to initialize, the job fails. This is the default value.</p>
Global Address	<p>Specifies whether the transform uses the Global Address engine for the Global Address Cleanse transform.</p> <p><i>Yes:</i> Activates the Global Address engine.</p> <p><i>No:</i> Disables the Global Address engine.</p>
USA	<p>Specifies whether the transform uses the USA engine for the Global Address Cleanse transform.</p> <p><i>Yes:</i> Activates the USA engine.</p> <p><i>No:</i> Disables the USA engine.</p>

Parent topic: [Global Address Cleanse \[page 552\]](#)

Related Information

[Report and Analysis \[page 554\]](#)

[Reference files \[page 558\]](#)

[Country ID options \(Global Address Cleanse\) \[page 559\]](#)

[Standardization Options: Country \[page 562\]](#)

[Standardization options \[page 563\]](#)

[Canada options: Global Address Cleanse \[page 572\]](#)
[Canada options: Report options \[page 575\]](#)
[Canada engine Suggestion List options \[page 576\]](#)
[Global Address \[page 578\]](#)
[USA: Global Address Cleanse \[page 589\]](#)
[Suggestion List: Global Address Cleanse \[page 592\]](#)
[Global Address Cleanse fields \[page 597\]](#)
[Global Address Cleanse sample configurations \[page 638\]](#)

8.5.9.5 Standardization Options: Country

Select a country name to determine the address standards the Global Address Cleanse transform uses.

Option	Description
Country Name	<p>Specify the country for which your standardization option settings apply.</p> <ul style="list-style-type: none">• Global (Apply to all countries): Apply global standardization to all source records.• <Specific Country> name: Choose a specific country from the dropdown list to apply the country-specific standardization to all source records.

Parent topic: [Global Address Cleanse \[page 552\]](#)

Related Information

[Report and Analysis \[page 554\]](#)
[Reference files \[page 558\]](#)
[Country ID options \(Global Address Cleanse\) \[page 559\]](#)
[Engines \[page 561\]](#)
[Standardization options \[page 563\]](#)
[Canada options: Global Address Cleanse \[page 572\]](#)
[Canada options: Report options \[page 575\]](#)
[Canada engine Suggestion List options \[page 576\]](#)
[Global Address \[page 578\]](#)
[USA: Global Address Cleanse \[page 589\]](#)
[Suggestion List: Global Address Cleanse \[page 592\]](#)
[Global Address Cleanse fields \[page 597\]](#)
[Global Address Cleanse sample configurations \[page 638\]](#)

8.5.9.6 Standardization options

Set Standardization options to standardize address data based on the country that you select, or for all countries.

The Options group includes all the options to standardize your address data. These settings apply to the country that you specify for ► [Standardization Options](#) ► [Country](#) ► [Country Name](#) ►.

Set these options once to apply to all source records, or duplicate the options group to specify a group of options for individual countries.

This option group is required.

Option	Description
Address Line Alias	<p>Specifies how to standardize the address line when the address contains alias address elements. Engines supported: Canada, Global Address, and USA.</p> <p><i>Convert</i>: Converts address lines based on official address line components instead of delivery address line components.</p> <p><i>Preserve</i>: Retains non postal preferred data in address lines unless the data is incorrect.</p> <div><p>i Note</p><p>Since December 31, 2013, Korea does not officially support the old Korean address system. The Korea reference data contains the old addresses as aliases to the new addressing system. Best practice is that you convert South Korean land-lot addresses to their new road name addresses. Set the Address Line Alias and Assign Locality options to <i>Convert</i>.</p></div>
Assign Locality	<p>Specifies how to standardize the locality name.</p> <p><i>Convert</i>: Converts the locality name to the locality name preferred by the postal authority for the country.</p> <p><i>Preserve</i>: Preserves the input locality name unless it is incorrect.</p> <p><i>Valid</i>: Retains the input locality name unless it is not valid for mailing. If it is not valid for mailing, replaces it with the preferred locality name.</p> <div><p>i Note</p><p>The Translate Major Locality option setting in the ► Global Address ► Country ► section takes precedence over the setting in this option, Assign Locality.</p></div> <div><p>i Note</p><p>Since December 31, 2013, Korea does not officially support the old Korean address system. The Korea reference data contains the old addresses as aliases to the new addressing system. Best practice is that you convert South Korean land-lot addresses to their new road name addresses. Set the Address Line Alias and Assign Locality options to <i>Convert</i>.</p></div>

Option	Description
Capitalization	<p>Specifies the casing of your address data.</p> <p><i>Mixed</i>: Converts data to initial capitals. For example, "MAIN STREET SOUTH" becomes "Main Street South."</p> <p><i>Upper</i>: Converts data to full capitals. For example, "Main Street South" becomes "MAIN STREET SOUTH."</p> <div> <p>i Note</p> <p>If you want consistent casing for your data, make sure that this option and the <i>Capitalization</i> setting in the Data Cleanse transform are the same.</p> </div>
Character Width Style	<p>Specifies whether to standardize half-width and full-width characters. This option applies only to Chinese, Japanese, and Korean data.</p> <p><i>Normal Width</i>: Converts full-width Latin characters to half-width and converts half-width Chinese, Japanese, and Korean characters to full width.</p> <p><i>Half Width</i>: Converts all characters to half width.</p> <p><i>Full Width</i>: Converts all characters to full width.</p>
Correct Assigned Data	<p>Specifies whether to use the parsed or corrected data for the assigned output fields of type Best.</p> <p><i>Yes</i>: Populates the Best components with corrected data.</p> <p><i>No</i>: Populates the Best components with parsed data.</p> <div> <p>i Note</p> <p>If you choose <i>No</i> for this option, the <i>Capitalization</i> option is the only available Standardization option for your assigned data.</p> </div>
Correct Unassigned Data	<p>Specifies whether the Global Address Cleanse transform standardizes your unassigned data.</p> <p><i>Yes</i>: Populates the Best components with corrected data.</p> <p><i>No</i>: Populates the Best components with parsed data.</p> <div> <p>i Note</p> <p>If you choose <i>No</i> for this option, the <i>Capitalization</i> option is the only available Standardization option for your unassigned data.</p> </div>

Option	Description
Country Style	<p>Specifies the standard to use to standardize country data.</p> <p><i>ISO_2CHAR</i>: Standardizes country data to the two-character ISO code, such as AU, CA, or US.</p> <p><i>ISO_3CHAR</i>: Standardizes country data to the three-character ISO code, such as AUS, CAN, or USA</p> <p><i>ISO_3DIGIT</i>: Standardizes country data to the three-digit ISO code, such as 038, 124, or 840.</p> <p><i>Name</i>: Standardizes country data to the full country name, such as Australia, Canada, or United States.</p> <p><i>Preserve</i>: Attempts to retain the country data from the input record, otherwise uses the corrected country value on output.</p>
Directional Punctuation	<p>Specifies whether to use punctuation in the abbreviated directional data.</p> <p><i>Yes</i>: Outputs directionals with punctuation. For example, N. or S.W..</p> <p><i>No</i>: Outputs directionals without punctuation. For example, N, SW.</p> <p><i>Preserve</i>: If there is punctuation in the input parsed component, returns the corrected punctuation if the transform does not find the input in the dictionary. For example, N/W on input is updated to N.W..</p>
Directional Style	<p>Specifies whether to abbreviate directional data.</p> <p><i>Long</i>: Uses fully spelled directionals such as "North," "South," "East," "West."</p> <p><i>Preserve</i>: Preserves the style used in the input record.</p> <p><i>Short</i>: Uses abbreviated directionals such as "N," "S," "E," "W."</p>
European Postcode Prefix	<p>Adds the one- to three-character European Postcode prefix, followed by a dash, for mail generated and distributed inside Europe.</p> <p><i>Yes</i>: Adds the European Postcode prefix.</p> <p><i>No</i>: Does not add the European Postcode prefix.</p> <p><i>Preserve</i>: If the transform finds one on input, retains the European Postcode prefix.</p>

❖ Example

In the following address, the D- is the European Postcode extension:

Hallesches Ufer 32-38
D-10963 Berlin
Germany

i Note

The European Postcode prefix is for mail distributed from one European country to another European country.

Option	Description
Extra Lines	<p>Specifies what to do with extra lines of non address data.</p> <p><i>Preserve</i>: Attempts to retain the extra line of non address data in the general location in which it was input.</p> <p><i>Remove</i>: Does not include any extra line of non address data in the standardized lines or multiline fields.</p> <p><i>Preferred</i>: Places all populated Extra fields above or below the multiline fields and standardized input lines based on the country data being processed. For example, Extra fields for Japan are located below the standardized lines.</p>
Format Assigned Data	<p>Specifies whether to format your assigned data based on the preferred address format of the country. For example, the format for Germany is:</p> <pre>{Primary_Name1} {Primary_Number} {Postcode1} {Locality} {Country}</pre> <ul style="list-style-type: none"> • <i>Yes</i>: Formats the assigned data. • <i>No</i>: Does not format the assigned data and leaves it in the location in which it was input. If data is added to the record, this data is placed based on the format string.
Format Unassigned Data	<p>Specifies whether to format your unassigned data based on the preferred address format for the country. For example, the format for Germany is:</p> <pre>{Primary_Name1} {Primary_Number} {Postcode1} {Locality} {Country}</pre> <ul style="list-style-type: none"> • <i>Yes</i>: Formats the unassigned data. • <i>No</i>: Does not format the unassigned data and leaves it in the line in which it was input.
Convert_Latin_Output_To_US_ASCII	<p>For Latin script records, if a character conversion is available, converts any extended ASCII characters in the Best component to US ASCII characters. The transform leaves any extended ASCII characters for which there is no conversion as is. For example, the transform leaves the degree symbol or inverted exclamation and question marks as is.</p> <div> <p>❖ Example</p> <p>With the input street name “Østerbrogade”, preserve the local character or convert it to the international data format “Osterbrogade” in the cleansed output.</p> </div> <ul style="list-style-type: none"> • <i>Yes</i>: Converts extended ASCII characters. • <i>No</i>: Does not convert extended ASCII characters. This is the default setting.
Include Country	<p>Specifies whether to include country names in standardized lines or multiline fields.</p> <ul style="list-style-type: none"> • <i>Yes</i>: Includes country name. • <i>No</i>: Does not include country name. • <i>Preserve</i>: If the transform finds the country name on input, retains the country name.

Option	Description
Include Locality Addition	<p>Specifies whether the <Locality1_Full> output field contains both the <Locality1_Name> and the <Locality1_Addition> information.</p> <ul style="list-style-type: none"> • Yes: Includes both the locality and the locality addition information. • No: Does not include the locality addition. • Preserve: If the transform finds a locality addition on input, it includes the locality addition. This is the default setting. <div> <p>i Note</p> <p>The <i>Locality Name Style</i> option in the Global Address Cleanse transform overrides this option. If the <i>Locality Name Style</i> option is set to <i>Short</i>, the <Locality1_Full> field does not contain the locality addition information.</p> </div> <div> <p>i Note</p> <p>If the <i>Translate Major Locality</i> option is set to translate the <Locality1> output field, the transform does not output the locality addition regardless of the setting in the <i>Include Locality Addition</i> option.</p> </div>
Include Unused Address Line Data	<p>Specifies whether to output the unused address line data for standardized lines and multiline fields. This option affects unused address data classified as remainder. This option does not affect unused address data classified as extra.</p> <ul style="list-style-type: none"> • Yes: Outputs the unused address line data in the remainder fields <Address_Line_Remainder1> through <Address_Line_Remainder4>. For example, 100 Main St Red House. • No: Does not output the unused address line data. For example, 100 Main St.
Include Unused Lastline Data	<p>Specifies whether to output the unused last line data for standardized lines and multiline fields:</p> <ul style="list-style-type: none"> • Yes: Outputs the unused last line data. • No: Does not output the unused last line data.

Option	Description																		
Locality Name Style	<p>Specifies the format for locality data in the <code><Locality1_Name></code> output field for addresses.</p> <p>This option applies to German addresses.</p> <ul style="list-style-type: none"><i>Preserve</i>: Preserves the locality data format as it was input. This is the default setting.<i>Short</i>: If the transform finds the locality data in the reference data, it outputs the locality data in the abbreviated version. <div><p>Note</p><p>To use the short locality name style, set the <i>Address Line Alias</i> option to <i>Convert</i>.</p></div> <div><p>Note</p><p>This option overrides the <i>Include Locality Addition</i> option.</p></div> <div><p>Note</p><p>If the <i>Translate Major Locality</i> option is set to translate the <code><Locality1></code> output field, the transform does not output short locality regardless of the <i>Locality Name Style</i> option setting.</p></div>																		
Move Multiline Data	<p>Determines the position of blank lines in output addresses.</p> <ul style="list-style-type: none"><i>Bottom</i>: Moves blank lines to the top and shifts the data to the bottom of the address block.<i>No</i>: Does not rearrange any lines, blank or otherwise.<i>Top</i>: If there are any blank lines, the transform moves them to the bottom of the address block and shifts the data to the top of the block. <div><p>❖ Example</p><p>Top:</p><div><table><tr><th></th><th>Input data:</th><th>Result of moving:</th></tr><tr><td>Line1</td><td>100 Market Street</td><td>Sycamore Building</td></tr><tr><td>Line2</td><td>Suite 202</td><td>Suite 202</td></tr><tr><td>Line3</td><td>Sycamore Building</td><td>100 Market St</td></tr><tr><td>Line4</td><td></td><td>Boston MA 02109</td></tr><tr><td>Line5</td><td>Boston MA 02109</td><td></td></tr></table></div></div>		Input data:	Result of moving:	Line1	100 Market Street	Sycamore Building	Line2	Suite 202	Suite 202	Line3	Sycamore Building	100 Market St	Line4		Boston MA 02109	Line5	Boston MA 02109	
	Input data:	Result of moving:																	
Line1	100 Market Street	Sycamore Building																	
Line2	Suite 202	Suite 202																	
Line3	Sycamore Building	100 Market St																	
Line4		Boston MA 02109																	
Line5	Boston MA 02109																		

Option	Description
Output Country Language	<p>Specify which language and script to use on output for the country name, not the entire record.</p> <ul style="list-style-type: none"> • <i>Preserve</i>: Preserves country name as it was on input. • <i>Catalan - Latin</i> • <i>Chinese - Hani</i> • <i>Danish - Latin</i> • <i>Dutch - Latin</i> • <i>English - Latin</i> • <i>Finnish - Latin</i> • <i>French - Latin</i> • <i>Greek - Greek</i> • <i>German - Latin</i> • <i>Hungarian - Latin</i> • <i>Italian - Latin</i> • <i>Japanese - Hani</i> • <i>Japanese - Kana</i> • <i>Korean - Hang</i> • <i>Norwegian - Latin</i> • <i>Polish - Latin</i> • <i>Portuguese - Latin</i> • <i>Spanish - Latin</i> • <i>Swedish - Latin</i>
Postal Phrase Punctuation	<p>If you choose <i>Short</i> for the <i>Postal Phrase Style</i> option, this option specifies whether to use punctuation in the postal abbreviation.</p> <ul style="list-style-type: none"> • <i>Yes</i>: Includes punctuation in postal abbreviations. For example, "P.O. Box". • <i>No</i>: Does not insert any punctuation for postal abbreviations. For example, "PO Box". • <i>Preserve</i>: If there is punctuation in the input parsed component, returns the corrected punctuation when the input is not found in the dictionary.
Postal Phrase Style	<p>Specifies whether to abbreviate postal phrases.</p> <ul style="list-style-type: none"> • <i>Long</i>: Outputs the fully spelled out postal phrase. For example, Post Office Box. • <i>Preserve</i>: If the transform finds a postal phrase on input, it retains the style of the postal phrase. • <i>Short</i>: Outputs the abbreviated postal phrase. For example, PO Box. The transform uses the setting in the <i>Postal Phrase Punctuation</i> option to determine the punctuation for this option.
Primary Type Punctuation	<p>If you choose <i>Short</i> for the <i>Primary Type Style</i> option, this option specifies whether to use punctuation in primary type abbreviations.</p> <ul style="list-style-type: none"> • <i>Yes</i>: Includes a period at the end of primary type abbreviations. For example, "St.". • <i>No</i>: Does not insert any punctuation at the end of primary type abbreviations. For example, "St". • <i>Preserve</i>: If there is punctuation in the input parsed component, returns the corrected punctuation if the transform does not find the input in the dictionary.

Option	Description
Primary Type Style	<p>Specifies the style for primary type address elements.</p> <ul style="list-style-type: none"> • Long: Uses fully spelled primary types such as Street, Avenue, Road, or Strasse. • Preserve: Retains the style used in the input record. • Short: Uses abbreviated primary type such as St, Ave, Rd, or Str. The transform determines the punctuation for this option by the setting in the <i>Primary Type Punctuation</i> option.
Region Style	<p>Specifies whether to abbreviate the region name. For example, state or province.</p> <ul style="list-style-type: none"> • Long: Uses the fully spelled region name. For example, California or Ontario. • Preserve: Retains the style used in the input record. • Short: Abbreviates the region name. For example, CA or ON.
Remove Address Apostrophes	<p>Specifies whether to include apostrophes in certain street names that include a DE L' or D'.</p> <ul style="list-style-type: none"> • Yes: Retains apostrophes in street names if it was present on input, for example, Rue D'Abbeville. • No: Removes apostrophes in street names, for example, Rue D Abbeville.
Secondary Description Punctuation	<p>If you choose Short for the <i>Secondary Description Style</i> option, the <i>Secondary Description Punctuation</i> option specifies whether to use punctuation in the abbreviation.</p> <ul style="list-style-type: none"> • Yes: Uses punctuation in the abbreviation, for example, "Apt". • No: Does not use punctuation, for example, "Apt". • Preserve: If there is punctuation in the input parsed component, returns the corrected punctuation if the transform does not find the input in the dictionary.
Secondary Description Style	<p>Specifies whether to abbreviate the secondary description, such as a unit or an apartment.</p> <ul style="list-style-type: none"> • Long: Uses the fully spelled secondary description. For example, "Apartment". • Preserve: Retains the style used in the input record. • Short: Abbreviates the secondary description. For example, "Apt". The transform uses the setting in the <i>Secondary Description Punctuation</i> option to determine the punctuation for this option.
Secondary Number Style	<p>Specifies the format of the secondary number such as a suite or apartment number. This option applies to Canada and New Zealand addresses.</p> <ul style="list-style-type: none"> • Attached: Converts all secondary ranges to the attached format, so that the secondary number is prepended to the primary number and separated with a delimiter. For example, for Canada addresses, places a dash between the secondary and primary range: 5-100 Main St. • Preserve: Preserves the style of the address as it was input. • Unattached: Converts all secondary ranges to the unattached format. For example, for Canada, it places the unit designator at the end of the primary address: 100 Main St Suite 5.

Option	Description
Street Name Style	<p>Specifies the format for street data for addresses.</p> <p>This option applies to addresses in Germany and the Netherlands.</p> <ul style="list-style-type: none"> Preserve: Preserves the street data format as it was input. <div> <p>Example</p> <p>For example:</p> <p>Annelien Kappeyne Van de Coppellostr 2 Herten Limburg 6049 HD</p> </div> <ul style="list-style-type: none"> Short: Outputs street data in the format preferred by the postal authority. For the Netherlands, the Short option returns a street address with a maximum of 24 characters in mixed case. <div> <p>Example</p> <p>For example:</p> <p>A K vd Coppellostr 2 Herten Limburg 6049 HD</p> </div> <div> <p>Note</p> <p>To use the short street name style, set the <i>Address Line Alias</i> option to <i>Convert</i>.</p> </div> <div> <p>Note</p> <p>The <i>Capitalization</i> option in the Global Address Cleanse transform overrides the setting for the <i>Street Name Style</i> option.</p> </div>
Use Local Primary Type Style	<p>Specifies whether to use the type style for primary address components that is present in the address data. Setting this option to <i>Yes</i> ignores the <i>Primary Type Style</i> option.</p> <p>This option applies to Austria, Germany, and Switzerland.</p> <ul style="list-style-type: none"> Yes: Uses the Primary Type Style that is present in the address data. No: Uses the Primary Type Style specified in the Primary Type Style option.
Use Postal Country Name	<p>Specifies which country data is output for countries that receive their postal service from another country. For example, if you use the USA engine and have addresses from the U.S. territories, the <i><Country></i> field is populated with the postal country (United States) rather than the territory name (such as American Samoa, Puerto Rico, and so on). The style of the <i><Country></i> field is still based on the Country Style option.</p> <p>If there is no postal country, the setting for this option does not change the output.</p> <ul style="list-style-type: none"> Yes: Uses the postal country name. No: Uses the territory country name.

Parent topic: [Global Address Cleanse \[page 552\]](#)

Related Information

[Report and Analysis \[page 554\]](#)

[Reference files \[page 558\]](#)

[Country ID options \(Global Address Cleanse\) \[page 559\]](#)

[Engines \[page 561\]](#)

[Standardization Options: Country \[page 562\]](#)

[Canada options: Global Address Cleanse \[page 572\]](#)

[Canada options: Report options \[page 575\]](#)

[Canada engine Suggestion List options \[page 576\]](#)

[Global Address \[page 578\]](#)

[USA: Global Address Cleanse \[page 589\]](#)

[Suggestion List: Global Address Cleanse \[page 592\]](#)

[Global Address Cleanse fields \[page 597\]](#)

[Global Address Cleanse sample configurations \[page 638\]](#)

8.5.9.7 Canada options: Global Address Cleanse

Set processing options in the Canada group of options for cleansing addresses from Canada.

Option	Description
Disable Certification	<p>Specify whether to disable Canada address certification.</p> <ul style="list-style-type: none">• Yes: The transform processes Canada addresses without including certification, and also processes with non POC (Point of Call) directories for non mailing purposes. The transform does not print the SERP report. Any list that you create with certification disabled cannot be used for mailing. This is the default setting.• No: The transform processes Canada addresses using processes required for SERP certification. This includes using POC directories and generating the SERP report. Any list that you create with certification enabled can be used for mailing.

Note

If you set this option to **Yes**, also set the following certification options to **Yes**:

- Enable LVR Rule
- Enable Rural Rule
- Postcode Only Search
- Postcode No Match Search

Option	Description
Dual Address	<p>Specify the transform behavior when there is a dual address.</p> <ul style="list-style-type: none"> • Position: Selects an address based on the arrangement of the input data. Tries to validate the address that is closest to the lower left corner of the address block. The closest address might be the postal or the street address depending on how the data was entered. Select Position to qualify for SERP certification. • Postal: Tries to validate based on the postal address. If that fails, tries to validate again based on the street address. • Street: Tries to validate based on the street address. If that fails, tries to validate again based on the postal address, such as rural route or PO box.
Enable LVR Rule	<p>Specify whether to enable the Canada Post LVR (Large Volume Receiver) postal code rule. The rule states that the LVR postal code is valid and cannot be changed to match other address components. Canada Post permits correction of LVR postal codes only when a unique address can be determined without changing the postal code.</p> <ul style="list-style-type: none"> • Yes: Regards any LVR address as assigned, even when the address line is so flawed that a match to the postal directory is impossible. Select Yes to qualify for SERP certification. • No: Disables the LVR postal code rule. Reports LVR addresses as unassigned when the address line is flawed.
Enable Rural Rule	<p>Specify whether to enable the Canada Post Rural rule. The rule states that any address with a valid rural postal code is considered valid. Rural postal codes have a zero in the second position.</p> <ul style="list-style-type: none"> • Yes: Regards any rural address as valid, even if the address line is so flawed that a match to the postal directory is impossible. Select Yes to qualify for SERP certification. • No: Reports a rural address as invalid if the address line is bad. <p>Canada Post states that you should not correct rural postal codes. However, the transform always attempts to correct all other address components except for the rural postal code. The rule applies under the following circumstances:</p> <ul style="list-style-type: none"> • When the address line is empty or contains bad data. • When the address does not have a postal code or it has an incorrect postal code. • When the locality (city) has just one postal code associated with it that is a rural postal code.
Output Address Language	<p>Specify the language for the output records.</p> <ul style="list-style-type: none"> • Convert: Uses French for records in Quebec, and English for records in other regions or provinces. • English: Converts output records to English. • French: Converts output records to French. • Preserve: Detects the input language and preserves that language upon output, ignoring the region or province. This is the default setting.
Parse Only	<p>Specify whether the transform should only parse input records.</p> <ul style="list-style-type: none"> • Yes: Parses records into discrete components, but does not perform a lookup in the postal directories. Parse Only is fast, but parsing results are unverified. • No: Parses records into discrete components and performs a lookup in the postal directories. Setting this option to No may slow down processing, but parsing results are verified when the appropriate reference data is available.

Option	Description
Postcode No Match Search	<p>Specify whether the transform should search the postal directories when the address line can be assigned, but it doesn't match the incoming postal code.</p> <p>Yes: Enable this option. Select Yes to qualify for SERP certification. If enabled, the transform searches the postal directories to determine if the following circumstances exist:</p> <ul style="list-style-type: none"> • If the incoming address line is a PO Box address, the postal code cannot be a valid postal code for an LVR (Large Volume Receiver), firm, or a civic (street) address, such as 100 Main St. • If the incoming address line is a civic (street) address, the postal code cannot be a valid postal code for an LVR PO Box address. <p>SERP rules state that, if either one of these circumstances exists, Data Services cannot assign the address.</p> <p>No: Disables this option. This is the default setting.</p> <p>A postal-code-only search in Data Services is time consuming. Therefore, disabling this search could improve your processing time.</p>
Postcode Only Search	<p>Specifies whether the transform should search for the address based on postal code only, and attempt to find a street record that contains the range.</p> <ul style="list-style-type: none"> • Yes: Enables the option. Select Yes to qualify for SERP certification. • No: Disables the option. In some cases, the result is a better address line. In other cases, the transform more reliably detects that it cannot assign an address line. <p>This option affects assignment when the input address line is badly incomplete. For example, when the address includes a range but no street name. When you set this option to Yes and the transform finds only one street record that contains the range, the transform assigns the address line from the postal code.</p>
Postcode Priority Over Street	<p>Specifies whether the transform weighs the postcode and street equally or whether street has priority over postcode. This option setting is important when the transform is trying to break a tie between two possible assignments:</p> <ul style="list-style-type: none"> • A near match on address line. • An exact match on postal code. <p>Yes: Assigns by placing more weight on the address line when breaking a tie. This assignment makes sense when you consider how common data-entry errors are for postal codes.</p> <p>No: Assigns by placing equal weight on the address line and the postal code when breaking a tie. Select No to qualify for SERP certification.</p>
Unit Description	<p>Specifies the unit description in English:</p> <ul style="list-style-type: none"> • Apartment: Uses Apartment as the default unit designator. • Default: Uses the default unit designator. • Unit: Uses "Unit" as the default unit designator.
Use Firm To Assign	<p>Specifies whether the firm is used to make an assignment and is displayed in a suggestion list.</p> <ul style="list-style-type: none"> • Yes: Uses and displays the firm. This is the default. • No: Does not use or display the firm.

Parent topic: [Global Address Cleanse \[page 552\]](#)

Related Information

[Report and Analysis \[page 554\]](#)

[Reference files \[page 558\]](#)

[Country ID options \(Global Address Cleanse\) \[page 559\]](#)

[Engines \[page 561\]](#)

[Standardization Options: Country \[page 562\]](#)

[Standardization options \[page 563\]](#)

[Canada options: Report options \[page 575\]](#)

[Canada engine Suggestion List options \[page 576\]](#)

[Global Address \[page 578\]](#)

[USA: Global Address Cleanse \[page 589\]](#)

[Suggestion List: Global Address Cleanse \[page 592\]](#)

[Global Address Cleanse fields \[page 597\]](#)

[Global Address Cleanse sample configurations \[page 638\]](#)

8.5.9.8 Canada options: Report options

Set the report options so that the software produces the Statement of Address Accuracy report information for SERP certification.

Complete the options in this group when you prepare a mailing list for Canada Post certification.

Option	Description
Customer Company Name	Specifies the company name of the organization for whom you are preparing the mailing (up to 40 characters).
Mailer Address1	Specifies the name and address of the person or organization for whom you are preparing the mailing (up to 40 characters per line).
Mailer Address2	
Mailer Address3	
Mailer Address4	
Customer CPC Number	Specifies the customer CPC number that is located on the Canada Post Contract (up to 15 characters).

Parent topic: [Global Address Cleanse \[page 552\]](#)

Related Information

[Report and Analysis \[page 554\]](#)
[Reference files \[page 558\]](#)
[Country ID options \(Global Address Cleanse\) \[page 559\]](#)
[Engines \[page 561\]](#)
[Standardization Options: Country \[page 562\]](#)
[Standardization options \[page 563\]](#)
[Canada options: Global Address Cleanse \[page 572\]](#)
[Canada engine Suggestion List options \[page 576\]](#)
[Global Address \[page 578\]](#)
[USA: Global Address Cleanse \[page 589\]](#)
[Suggestion List: Global Address Cleanse \[page 592\]](#)
[Global Address Cleanse fields \[page 597\]](#)
[Global Address Cleanse sample configurations \[page 638\]](#)

8.5.9.9 Canada engine Suggestion List options

Set these options when you want to generate suggestion lists for your Canada address data.

Option	Description
Address Lines Match Minimum	<p>Specifies the similarity score required for address-line suggestions. This score then determines which suggestions will be returned in the list. A higher number indicates that the suggestion must be more similar to the input in order to be returned as a possible suggestion.</p> <p>Type a value from 0 to 80.</p>
Address Range	<p>Specifies a span around the input primary address range for which to return suggestions. By using this option, you can limit the suggestions returned to be within a few blocks of your input. For example, assume you entered 500 for this value. Then, you submit the following street address:</p> <p>1000 Pine St.</p> <p>Suggestions would be returned in a range from 750 to 1250 Pine Street.</p> <p>If you don't want to limit the ranges returned in suggestions, type 0.</p>

Option	Description
Combine Overlapping Ranges	<p>Specifies whether individual suggestions with overlapping ranges are combined.</p> <p>Yes: Ignores gaps and overlaps in ranges.</p> <p>Set this option to Yes if you want to limit the number of total suggestions presented to your user. However, you might not see gaps of invalid ranges that would be apparent if this option was set to No.</p> <p>For example, the following suggestions would be presented if this option is set to No:</p> <p>1000-1099 Maple Ave</p> <p>1100-1199 Maple Ave</p> <p>But this suggestion would only show if set to Yes:</p> <p>1000-1199 Maple Ave</p> <p>No: Does not combine overlapping ranges.</p>
Enable Suggestion Lists	<p>Specifies whether suggestion lists are generated.</p> <p>Yes: Generates suggestion lists when assignment candidates are present.</p> <p>No: Does not generate suggestion lists.</p>
Lastlines Match Minimum	<p>Specifies the similarity score required for lastline suggestions. This score then determines which suggestions will be returned in the list. A higher number indicates that the suggestion must be more similar to the input in order to be returned as a possible suggestion.</p> <p>Type a value from 0 to 80.</p>
Max Number Address Lines	<p>Specifies the maximum number of address line suggestions that can be generated.</p> <p>You might set this option in order to limit the size of the SOAP documents being sent by the web service, or to limit the maximum number of suggestions that your users would have to choose from. However, by setting a maximum, you may occasionally eliminate a suggestion from the list that could be the correct one.</p> <p>The minimum number you can enter is 2. The maximum number you can enter is 50.</p>
Max Number Lastlines	<p>Specifies the maximum number of lastline suggestions that can be generated.</p> <p>You might set this option in order to limit the size of the SOAP documents being sent by the web service, or to limit the maximum number of suggestions that your users would have to choose from. However, by setting a maximum, you may occasionally eliminate a suggestion from the list that could be the correct one.</p> <p>The minimum number you can enter is 2. The maximum number you can enter is 15.</p>

Parent topic: [Global Address Cleanse \[page 552\]](#)

Related Information

[Report and Analysis \[page 554\]](#)

[Reference files \[page 558\]](#)
[Country ID options \(Global Address Cleanse\) \[page 559\]](#)
[Engines \[page 561\]](#)
[Standardization Options: Country \[page 562\]](#)
[Standardization options \[page 563\]](#)
[Canada options: Global Address Cleanse \[page 572\]](#)
[Canada options: Report options \[page 575\]](#)
[Global Address \[page 578\]](#)
[USA: Global Address Cleanse \[page 589\]](#)
[Suggestion List: Global Address Cleanse \[page 592\]](#)
[Global Address Cleanse fields \[page 597\]](#)
[Global Address Cleanse sample configurations \[page 638\]](#)

8.5.9.10 Global Address

The Global Address group includes options for specified countries, or for global addresses.

The Global Address group includes settings for reference files, country-specific options, suggestion list options, and report options.

[Global Address: Country Options \[page 579\]](#)

Set address assignment options for specified countries.

[Global Address: Suggestion List Options \[page 583\]](#)

Specify suggestion list options for global addresses.

[Global Address: Report Options \[page 586\]](#)

Use the *Report Options* to add required certification information to reports for Australia and New Zealand.

[Report options for New Zealand \[page 587\]](#)

Set options to include the required New Zealand Statement of Accuracy (SOA) information to reports.

[Report options for Australia \[page 588\]](#)

Set options to include the required Australia AMAS (Address Matching Processing Summary) information to reports.

Parent topic: [Global Address Cleanse \[page 552\]](#)

Related Information

[Report and Analysis \[page 554\]](#)
[Reference files \[page 558\]](#)
[Country ID options \(Global Address Cleanse\) \[page 559\]](#)
[Engines \[page 561\]](#)
[Standardization Options: Country \[page 562\]](#)

[Standardization options \[page 563\]](#)
[Canada options: Global Address Cleanse \[page 572\]](#)
[Canada options: Report options \[page 575\]](#)
[Canada engine Suggestion List options \[page 576\]](#)
[USA: Global Address Cleanse \[page 589\]](#)
[Suggestion List: Global Address Cleanse \[page 592\]](#)
[Global Address Cleanse fields \[page 597\]](#)
[Global Address Cleanse sample configurations \[page 638\]](#)

8.5.9.10.1 Global Address: Country Options

Set address assignment options for specified countries.

Specify a country and then set address assignment options for the specified country.

Option	Description
<i>Options</i> group	
Retain Postcode if Valid Format	<p>Set this option for processing valid or invalid input postcodes.</p> <p><i>Yes:</i> Retains input postcode unless it is in an invalid format for the country and there is a single postcode match, or intelligent matching is possible.</p> <p><i>No:</i> Default setting. Updates output postcode if there is a single postcode match, or intelligent matching is possible. Otherwise retain input postcode.</p> <div> <p>i Note</p> <p>If the postcode is blank, the transform populates the postcode if it finds a single postcode match.</p> </div>
Dual Address	<p>Specifies the action to take when the Global Address engine encounters a dual address.</p> <p><i>Position:</i> Selects an address based on the arrangement of the input data. The transform attempts to assign the first address found in the input data. If it cannot assign the first address, it attempts to assign the next address it finds.</p> <p><i>Postal:</i> Attempts to validate based on the postal address. If that fails, attempts again based on the street address.</p> <p><i>Street:</i> Attempts to validate based on the street address. If that fails, attempts again based on the postal address.</p>

Option	Description
Disable Certification	<p>Specifies whether to perform non certified or certified processing of addresses for Australia, France, or New Zealand.</p> <div> <p>i Note</p> <p>If the reference data is missing or older than the certification requirements for any of these countries (Australia, France, or New Zealand), the software issues a warning message. If the software issues the message for data that is not required for the country you are processing, you can ignore the message. To avoid receiving this message, replicate the country options group for each country (Australia, France, and New Zealand) and set the <i>Disable Certification</i> option to <i>No</i> only for the country data you are processing. When you set the options for each country, the software uses all other options in the country-specific group in place of the options selected in the global group.</p> </div> <p>If the country is Australia:</p> <ul style="list-style-type: none"> • <i>Yes</i>: Enables non certified features and extends the directory expiration for non mailing purposes. Extend the directory expiration period for up to 14 months from the date the directories were created. Processing with expired directory data is allowed when you are not planning to use the records for AMAS mailing purposes. Processing with expired directory data is ideal for data warehousing industries, for example. However, when you select <i>Yes</i>, you cannot print the AMAS report. You cannot use any lists created with expired directories for postage discounts. Data directories expire after 15 months when certification is disabled. • <i>No</i>: Uses the most current directory information, disables non certified features, and enables printing of the AMAS report. <p>If the country is France:</p> <ul style="list-style-type: none"> • <i>Yes</i>: Enables non certified processing of France addresses for non mailing purposes. • <i>No</i>: Enables certified processing of France addresses for mailing purposes. The software removes all punctuation except for firm data. The software removes all accent characters (only A-Z and 0-9 are allowed). The returned address is in 6 lines.

Option	Description
	<div> <div>i Note</div> <p>To return the address in all upper case, set the Capitalization option under Standardization Options Country Options to <i>Upper</i>.</p> </div> <p>If the country is New Zealand:</p> <ul style="list-style-type: none"> Yes: Enables non certified processing of New Zealand addresses and allows processing with expired directories for non mailing purposes. When you select <i>Yes</i>, you cannot print the SOA Report. You cannot use any list that is created with certification disabled for mailing. No: Enables certified processing for New Zealand and enables printing of the SOA report.
Output Address Script	<p>Specifies how a native script is output:</p> <ul style="list-style-type: none"> Supports Chinese if the input character script is CJKK Supports Russian if the input character script is CYRL <p><i>Latin:</i> Transliterates the native script and outputs the data as Latin.</p> <p><i>Preserve:</i> Outputs the data in the native script.</p> <div> <div>i Note</div> <p>The <i>Translate Major Locality</i> option setting takes precedence over the <i>Output Address Script</i> option for the locality.</p> </div>
Translate Major Locality	<p>Specifies whether to translate the <code><Locality1></code> output field for major localities.</p> <p><i>English:</i> Translates the output to English, if available.</p> <p><i>Preserve:</i> Default value. Outputs the locality data as it is input.</p> <div> <div>i Note</div> <p>The <i>Translate Major Locality</i> option takes precedence over the <i>Assign Locality</i> option. It also takes precedence over the <i>Output Address Script</i> option for the locality.</p> </div>

Option	Description
Use Firm to Assign	<p>Specifies whether to use the firm name to make an assignment and to display the firm name in a suggestion list.</p> <p><i>Yes</i>: Default value. Uses and displays the firm.</p> <p><i>No</i>: Does not use or display the firm.</p>
Processing Timeout	<p>Specifies the timeout duration in milliseconds. After the duration is exceeded, the software stops processing and issues the information code 7000. You can use a substitution variable for this option.</p> <p><i>0</i>: Default value. Disables the timeout.</p> <p>Number greater than zero: Indicates the amount of time after which the software aborts processing.</p> <div> <p>i Note</p> <p>Best practice is to set the <i>Processing Timeout</i> at or above 100 milliseconds.</p> </div>
Suppress Suggestion List With no Range	<p>Specifies whether a suggestion list is generated when “S/N” or no range is found on input, and the only records found in the reference data are range records.</p> <p><i>Yes</i>: The software does not generate a suggestion list. The software returns the information code 3030 and makes a primary name level match assignment.</p> <p><i>No</i>: Default value. The software generates a suggestion list with the range values in the reference data record.</p>
<i>Country</i> block	
Country Name	<ul style="list-style-type: none"> <i>Global (Apply to all countries)</i>: All options under Country Options apply to all countries in your data. <i><Specific Country></i>: Choose a specific country from the dropdown list to apply Country Options to a specific country.

Parent topic: [Global Address \[page 578\]](#)

Related Information

[Global Address: Suggestion List Options \[page 583\]](#)

[Global Address: Report Options \[page 586\]](#)

[Report options for New Zealand \[page 587\]](#)

[Report options for Australia \[page 588\]](#)

8.5.9.10.2 Global Address: Suggestion List Options

Specify suggestion list options for global addresses.

There are suggestion list options in the USA and Canada blocks. The options that you set under Global Address are for global addresses.

Option	Description
Enable Suggestion Lists	<p>Specifies whether the transform generates suggestion lists for global addresses.</p> <p>Yes: Generates suggestion lists when assignment candidates are present.</p> <p>No: Does not generate suggestion lists.</p>

Option	Description
Enable Lastline Drilldown	<p>Specifies whether you can drill down to the next level of the lastline component in suggestion lists after selecting its parent component.</p> <div> <p>❖ Example</p> <p>For example, given an input country, suggestion list provides a list of available regions. After you select one of the regions, suggestion list provides a list of available locality1 options. After you select one of the localities, suggestion list provides a list of available locality2 options, and so on.</p> </div> <div> <p>❖ Example</p> <p>For example, for China, you select a region, then you select a city within the selected region, and then you select a district within the selected city.</p> </div> <p>Yes: Allows drilldown on lastlines.</p> <p>No: Does not allow drilldown on lastlines.</p> <div> <p>i Note</p> <p>To use this option, set the <i>Enable Suggestion Lists</i> option to <i>Yes</i>.</p> </div> <div> <p>i Note</p> <p>The <i>Enable Lastline Drilldown</i> option is supported by the Global Engine of the Global Address transform; it is not available for the Canada and USA engines.</p> </div> <p>More information: The expected input for lastline drilldown is country, and all other input fields are blank. For example, input fields locality, region, postcode, firm, multiline.</p> <div> <p>i Note</p> <p>As always, comply with one of the accepted formats for your input fields combination. Accepted formats are multiline, hybrid, and discrete.</p> </div> <p>For a few countries, such as India, localities are not always linked with region in the data directory. Therefore, the software may not return localities in the suggestion list. For countries and regions with many localities, the suggestion list may perform slower.</p> <p>A suggestion list using the <i>Enable Lastline Drilldown</i> option is the same as a typical lastline suggestion list, except that there are no postcode or address type components in the suggestion entry.</p> <p>For a list of supported countries and country codes, see Countries supported by lastline drilldown [page 802].</p>

Option	Description
Combine Overlapping Ranges	<p>Specifies whether individual suggestions with overlapping ranges are combined.</p> <p>Yes: Ignores gaps and overlaps in ranges.</p> <p>No: Does not combine overlapping ranges.</p> <p>You might set this option to Yes if you want to limit the number of total suggestions presented to your user. However, you might not see gaps of invalid ranges that would be apparent if this option was set to No.</p> <div> <p>❖ Example</p> <p>For example, a suggestion list might show the following suggestions if this option is set to No:</p> <p>1000-1099 Maple Ave 1100-1199 Maple Ave</p> <p>But would only show this suggestion if set to Yes:</p> <p>1000-1199 Maple Ave</p> </div> <div> <p>i Note</p> <p>If the option is set to No, best practice is to increase the size of the <code><Suggestion_List></code> output field so that it can contain all of the suggestions.</p> </div>
Address Range	<p>Specifies a span around the input primary address range for which to return suggestions. By using this option, you can limit the suggestions returned to within a few blocks of your input.</p> <div> <p>❖ Example</p> <p>For example, assume that you entered 500 for Address Range. Then, you submit the following street address: 1000 Pine St.</p> <p>The software would only return suggestions in a range from 750 to 1250 Pine Street.</p> </div> <p>If you don't want to limit the ranges returned in suggestions, type 0. The maximum value is 5000.</p>
Max Number Address Lines	<p>Specifies the maximum number of address line suggestions that can be generated.</p> <p>Minimum setting: 2</p> <p>Maximum setting: 10,000</p> <p>Default setting: 100</p> <p>You might set this option to limit the size of the SOAP documents being sent by the web service, or to limit the maximum number of suggestions that your users would have to choose from.</p>

Option	Description
Max Number Lastlines	<p>Specifies the maximum number of lastline suggestions that can be generated.</p> <p>Minimum setting: 2</p> <p>Maximum setting: 10,000</p> <p>Default setting: 60</p> <div> <p>❖ Example</p> <p>Set the <i>Max Number Lastlines</i> option to limit the size of SOAP documents sent by the web service. Or, set this option to limit the maximum number of suggestions that users have to choose from.</p> </div> <div> <p>i Note</p> <p>You cannot set this option when the <i>Enable Lastline Drilldown</i> is set to <i>Yes</i>.</p> </div>

Parent topic: [Global Address \[page 578\]](#)

Related Information

[Global Address: Country Options \[page 579\]](#)

[Global Address: Report Options \[page 586\]](#)

[Report options for New Zealand \[page 587\]](#)

[Report options for Australia \[page 588\]](#)

[Countries supported by lastline drilldown \[page 802\]](#)

8.5.9.10.3 Global Address: Report Options

Use the *Report Options* to add required certification information to reports for Australia and New Zealand.

Option	Description
Australia	Set report options to complete the required Australia AMAS (Address Matching Processing Summary) Report required by Australia postal authorities.
New Zealand	Set report options to complete the required Statement of Accuracy (SOA) report required by New Zealand postal authorities.

Parent topic: [Global Address \[page 578\]](#)

Related Information

[Global Address: Country Options \[page 579\]](#)

[Global Address: Suggestion List Options \[page 583\]](#)

[Report options for New Zealand \[page 587\]](#)

[Report options for Australia \[page 588\]](#)

8.5.9.10.4 Report options for New Zealand

Set options to include the required New Zealand Statement of Accuracy (SOA) information to reports.

Completing this group of options is only required when you are preparing a mailing that needs a New Zealand SOA report for postal certification.

Option	Description
Customer Number	Specify the New Zealand Post number assigned to the customer for whom you are preparing the list. New Zealand post requires the customer number in the SOA report to qualify for postage discounts.
Customer Company Name	Specify the customer company name. The customer company is the entity for which you are preparing the list. Maximum is 40 characters.
Mailer Address1 Mailer Address2 Mailer Address3 Mailer Address4 Mailer Address5 Mailer Address6	Specify the name and address of the person or organization for which you are preparing the list. Maximum is 29 characters per line. <div><div>❖ Example</div><div>Mailer Address1 = ABCD Mailing, Inc. Mailer Address2 = 123 Main St. Mailer Address3 = Johnsonville Mailer Address4 = Wellington 6004 Mailer Address5 = NEW ZEALAND</div></div>
SOA Issuer Name	Specify the company name that prepared the list. Maximum of 40 characters.
File Name	Specify the input file name. For example, <code>new_zealand.dbf</code> .

Parent topic: [Global Address \[page 578\]](#)

Related Information

[Global Address: Country Options \[page 579\]](#)

[Global Address: Suggestion List Options \[page 583\]](#)

[Global Address: Report Options \[page 586\]](#)

[Report options for Australia \[page 588\]](#)

8.5.9.10.5 Report options for Australia

Set options to include the required Australia AMAS (Address Matching Processing Summary) information to reports.

Completing this group of options is only required when you are preparing a mailing that needs an Australia AMAS report for postal certification.

Option	Description
Customer Company Name	Specify the customer company name. The customer company is the entity for which you are preparing the list. Maximum is 40 characters.
Mailer Address1 Mailer Address2 Mailer Address3 Mailer Address4	<div><p>Specify the name and address of the person or organization for which you are preparing the list. Maximum is 29 characters per line.</p><div><p>❖ Example</p><p>Mailer Address1 = ABCD Mailing, Inc. Mailer Address2 = 123 Main St. Mailer Address3 = West Beach SA 1234 Mailer Address4 = Australia</p></div></div>
List Name	Specify the name of your database or mailing list. Maximum is 40 characters. For example, enter the file name, your title, or a formal name for the list.
File Name	Specify the input file name. Maximum is 40 characters. For example, <code>australia.dbf</code> .

Parent topic: [Global Address \[page 578\]](#)

Related Information

[Global Address: Country Options \[page 579\]](#)

[Global Address: Suggestion List Options \[page 583\]](#)

[Global Address: Report Options \[page 586\]](#)

[Report options for New Zealand \[page 587\]](#)

8.5.9.11 USA: Global Address Cleanse

Use the USA options in the Global Address Cleanse transform to set processing options for the United States of America and territories.

Set options for address parsing and suggestion lists for addresses from the United States and territories.

[USA Options: Global Address Cleanse \[page 589\]](#)

Define options for standardizing United States and territories fields that are in your data.

[USA: Suggestion List Options \[page 591\]](#)

Set Suggestion List Options to enable suggestions for addresses from the United States and territories.

Parent topic: [Global Address Cleanse \[page 552\]](#)

Related Information

[Report and Analysis \[page 554\]](#)

[Reference files \[page 558\]](#)

[Country ID options \(Global Address Cleanse\) \[page 559\]](#)

[Engines \[page 561\]](#)

[Standardization Options: Country \[page 562\]](#)

[Standardization options \[page 563\]](#)

[Canada options: Global Address Cleanse \[page 572\]](#)

[Canada options: Report options \[page 575\]](#)

[Canada engine Suggestion List options \[page 576\]](#)

[Global Address \[page 578\]](#)

[Suggestion List: Global Address Cleanse \[page 592\]](#)

[Global Address Cleanse fields \[page 597\]](#)

[Global Address Cleanse sample configurations \[page 638\]](#)

8.5.9.11.1 USA Options: Global Address Cleanse

Define options for standardizing United States and territories fields that are in your data.

Option	Description
Parse Only	<p><i>Yes:</i> Parses records into discrete components, but does not perform a lookup in the postal directories. <i>Parse Only</i> is fast, but parsing results are unverified.</p> <p><i>No:</i> Parses records into discrete components and performs a lookup in the postal directories. Setting this option to <i>No</i> may slow down processing, but parsing results are verified.</p>

Option	Description
Unit Description	<p>Specify how to standardize unit descriptions such as apartment, suite, room, or floor.</p> <p><i>Convert</i>: Uses the unit description found in the postal directory.</p> <p><i>Preserve</i>: Preserves the unit description from the input record, correcting any spelling errors.</p>
Dual Address	<p>Specify the action to take when the transform encounters a dual address.</p> <p><i>Position</i>: Selects an address based on the arrangement of the input data.</p> <p>The transform attempts to validate the address that is closest to the lower left corner of the address block. That address might be the postal or the street address based on how the data was entered.</p> <p><i>Postal</i>: Attempts to validate the address based on the postal address. If that fails, attempts to validate the address again based on the street address.</p> <p><i>Street</i>: Attempts to validate the address based on the street address. If that fails, attempts to validate the address again based on the postal address, such as rural route or PO Box.</p>
Use Firm To Assign	<p>Specify whether the transform uses the firm to make an assignment and to display firm in a suggestion list.</p> <p><i>Yes</i>: Uses and displays the firm. Default setting.</p> <p><i>No</i>: Does not use or display the firm.</p>

Parent topic: [USA: Global Address Cleanse \[page 589\]](#)

Related Information

[USA: Suggestion List Options \[page 591\]](#)

[Standardization options \[page 563\]](#)

8.5.9.11.2 USA: Suggestion List Options

Set Suggestion List Options to enable suggestions for addresses from the United States and territories.

Option	Description
Enable Suggestion Lists	<p>Specifies whether the transform generates suggestion lists.</p> <p>Yes: Generates suggestion lists when assignment candidates are present.</p> <p>No: Does not generate suggestion lists.</p>
Max Number Lastlines	<p>Specifies the maximum number of lastline suggestions that can be generated.</p> <p>Minimum setting: 2</p> <p>Maximum setting: 100</p> <p>Default setting: 15</p> <div><p>❖ Example</p><p>Set the Max Number Lastlines option to limit the size of SOAP documents sent by the web service. Or, set this option to limit the maximum number of suggestions that users have to choose from.</p></div>
Max Number Address Lines	<p>Specifies the maximum number of lastline suggestions that can be generated.</p> <p>Minimum setting: 2</p> <p>Maximum setting: 100</p> <p>Default setting: 50</p> <div><p>❖ Example</p><p>Set the <i>Max Number Lastlines</i> option to limit the size of SOAP documents sent by the web service. Or, set this option to limit the maximum number of suggestions that users have to choose from.</p></div>
Lastlines Match Minimum	<p>Specifies the similarity score required for last line suggestions. This score then determines which suggestions are returned in the list.</p> <p>Type a value from <i>0</i> to <i>80</i>.</p> <p>A higher number indicates that the suggestion must be more similar to the input to be returned as a possible suggestion.</p>
Address Lines Match Minimum	<p>Specifies the similarity score required for address line suggestions. This value then determines which suggestions are returned in the list.</p> <p>Type a value from <i>0</i> to <i>80</i>.</p> <p>A higher number indicates that the suggestion must be more similar to the input to be returned as a possible suggestion.</p>

Option	Description
Combine Overlapping Ranges	<p>Specifies whether the transform combines individual suggestions with overlapping ranges.</p> <p>Yes: Ignores gaps and overlaps in ranges.</p> <p>No: Does not combine overlapping ranges.</p> <p>You might set this option to Yes if you want to limit the number of total suggestions presented to your user. However, you might not see gaps of invalid ranges that would be apparent if this option was set to No.</p> <p>For example, a suggestion list might show the following suggestions if this option is set to No:</p> <p>1000-1099 Maple Ave 1100-1199 Maple Ave</p> <p>But a suggestion list might only show the following suggestion if set to Yes:</p> <p>1000-1199 Maple Ave</p>
Address Range	<p>Specifies a span around the input primary address range for which to return suggestions. By using this option, you can limit the suggestions returned to be within a few blocks of your input.</p> <div> <p>❖ Example</p> <p>For example, assume that you entered 500 for this value. Then, you submit the following street address:</p> <p>1000 Pine St.</p> <p>The software would only return suggestions in a range from 750 to 1250 Pine Street.</p> </div> <p>If you don't want to limit the ranges returned in suggestions, type 0.</p>

Parent topic: [USA: Global Address Cleanse \[page 589\]](#)

Related Information

[USA Options: Global Address Cleanse \[page 589\]](#)

8.5.9.12 Suggestion List: Global Address Cleanse

Select the components and the format for the transform suggestion list string.

The Global Address Cleanse transform uses settings from each country suggestion list settings, and the settings that you make in this [Suggestion List Options](#) to determine the content of suggestion lists.

The following table contains descriptions for the three format options in the [Suggestion List](#) group.

Option	Description
Output Style	<p>Specify the format for the output suggestion list data.</p> <ul style="list-style-type: none"> Delimited: Outputs the suggestion list data in a delimited text format. Uses the delimiters that you specify in the <i>Delimiter</i> and <i>Field Delimiter</i> options. XML: Default setting. Outputs the suggestion list data as hierarchical XML. If you integrate suggestion lists via the web service, you may use this option. Use your own XML tools to parse the suggestion list data.
Delimiter	<p>Specify a character to use to separate each suggestion in a suggestion list. The transform considers this value only if the <i>Output Style</i> option is set to <i>Delimited</i>.</p> <p>The value can be any character or string. Common delimiters include a pipe symbol (), or a string of multiple asterisks (***). Select a value that is different from the <i>Field Delimiter</i> value.</p>
Field Delimiter	<p>Specify a character to use to separate each field in a single suggestion. The transform considers this value only if the <i>Output Style</i> option is set to <i>Delimited</i>.</p> <p>Each suggestion can be made up of one or more fields. If you choose to retrieve multiple fields per suggestion, the software separates the fields with the character specified here.</p> <p>This value can be any character or string. The default value is a pipe symbol (). Select a delimiter that is different from the <i>Delimiter</i> value.</p>

Parent topic: [Global Address Cleanse \[page 552\]](#)

Related Information

[Report and Analysis \[page 554\]](#)

[Reference files \[page 558\]](#)

[Country ID options \(Global Address Cleanse\) \[page 559\]](#)

[Engines \[page 561\]](#)

[Standardization Options: Country \[page 562\]](#)

[Standardization options \[page 563\]](#)

[Canada options: Global Address Cleanse \[page 572\]](#)

[Canada options: Report options \[page 575\]](#)

[Canada engine Suggestion List options \[page 576\]](#)

[Global Address \[page 578\]](#)

[USA: Global Address Cleanse \[page 589\]](#)

[Global Address Cleanse fields \[page 597\]](#)

[Global Address Cleanse sample configurations \[page 638\]](#)

8.5.9.12.1 Suggestion List Components: Global Address Cleanse

Select the components to include in all suggestion lists, global addresses, USA, and Canada.

For each component, select [Yes](#) or [No](#) based on the following information:

- Select [Yes](#) to include the component in suggestion lists.
- Select [No](#) to exclude the component from suggestion lists.

Component	Description
Selection	Returns a unique index number that distinguishes this suggestion from the other suggestions in the returned list. The suggestion selection number ranges from 1 to the number of suggestion selections in the suggestion list.
Locality1-4	Returns the city, town, or suburb. Additional locality information goes in Locality4.
Locality1-4_Official	Returns the locality name that is preferred by the postal authority.
Postcode	Returns the postal code.
Postcode1	Australia: Returns a four-digit postcode Canada: Returns the first three characters (FSA) of the postal code. Global: Returns the postal code. USA: Returns the five-digit primary postal code (ZIP Code). Does not include the four-digit secondary postal code (ZIP4).
Postcode2	Returns the secondary postal code. Canada: Returns the last three characters (LDU) of the postal code. USA: Returns the four-digit ZIP Code (ZIP4). The ZIP4 follows the primary postal code with a hyphen placed between, such as 54601-1234.
Region1-2	Returns the state, province, or region.
Primary Number Low Primary Number High	Returns the low and high primary number. If the house number is a range, such as 100-102: <ul style="list-style-type: none">• Primary Number Low = 100• Primary Number High = 102 If the house number is not a range, both fields contain the house number. For example: <ul style="list-style-type: none">• Primary Number Low = 100• Primary Number High = 100
Primary Number Description	Returns a description that precedes the primary number. For example, LOT (Australia).
Primary Number Extra	Returns data that is found near the parsed primary number, which cannot be identified or does not belong in a standardized address.

Component	Description
Primary Number Full	Returns the primary number, primary number description, and primary number extra.
Primary Prefix1-2	Returns the abbreviated or full directional that precedes a street name. For example, N, South, NW, or SE.
Primary Name1-4	Returns the street description. For example, <code><Primary_Name1></code> may return "Marina" and <code><Primary_Name2></code> may return "The Slipway."
Primary Type 1-4	Returns the primary name type. For example, rue, strasse, street, Ave, or Pl.
Primary Postfix1-2	Returns the abbreviated or full directional that follows a street name. For example, N, South, NW, SE.
Primary Name Full1 Primary Name Full2	Returns the primary name, primary type, primary prefix, and primary postfix.
Primary Name Full3 Primary Name Full4	Returns the primary name and primary type.
Delivery Installation Name	Returns the delivery installation city name, which is the same as the city name in most cases. If it is the same as the city name, the software omits it from the address line.
Delivery Installation Qualifier	Returns the delivery installation qualifier (for example, for the address RR2 Vancouver Stn Main, returns Main).
Delivery Installation Type	<p>Returns the delivery installation type.</p> <p>English:</p> <ul style="list-style-type: none"> • PO: Post Office • RPO: Retail Post Outlet • STN: Station • LCD: Letter Carrier Depot • CMC: Community Mail Center • CDO: Commercial Dealership Outlet <p>French:</p> <ul style="list-style-type: none"> • BDP: Bureau de Poste • CSP: Comptoir Service Postal • SUCC: Succursale • PDF: Poste de Facteurs • CPC: Centre Postal Communautaire • CC: Concession Commerciale

Component	Description
Primary Side Indicator	<p>Returns the valid primary side indicator. The value applies to Street and PO box.</p> <p>E: The record is even-numbered.</p> <p>O: The record is odd-numbered.</p> <p>B: The record covers both the even- and odd-numbered sides of the street and PO Boxes.</p>
Firm	Returns the firm, company, or organization name.
Building Name	<p>Returns the building name for the address, which in some countries is used in place of the primary number.</p> <div> <p>❖ Example</p> <p>Input address from the U.K.:</p> <p>White House, High Street</p> <p>For this address, "White House" is the building name instead of using a primary number such as "100 High Street".</p> </div> <p>In some cases, an address includes the building name and primary number.</p>
Unit Description	Returns the unit description within a building. For example, Room, Unit, Apt, Suite, and so on.
Unit Number Low Unit Number High	<p>Returns the low and high unit number if the unit number is a range. For example, with a unit number range of 1-20:</p> <ul style="list-style-type: none"> • Unit Number Low = 1 • Unit Number High = 20
Stairwell Description	Returns the entrance or stairwell identifier for a building. For example, Stiege.
Stairwell Name	Returns the name or number of an entrance or stairwell for a building. For example, Stiege "1."
Floor Number High Floor Number Low	<p>Returns the low and high number if the floor number is a range. For example, with the range 20-22:</p> <ul style="list-style-type: none"> • Floor Number Low = 20 • Floor Number High = 22 <p>If the floor number is not a range, both fields contain the floor number (for example:</p> <ul style="list-style-type: none"> • Floor Number Low = 20 • Floor Number High = 20
Floor Description	Returns the level description, such as Floor.

Component	Description
Secondary Side Indicator	Returns the secondary record side indicator. This value applies to floors and units: E : The secondary record is even-numbered. O : The secondary record is odd-numbered. B : The secondary record covers both the even and odd-numbered values.
Sugg Range Type	Indicates the range type for each suggestion: <ul style="list-style-type: none"> • PRIM = Suggestion represents a range of house numbers. • UNIT = Suggestion represents a range of unit numbers. • FLOOR = Suggestion represents a range of floor numbers. • blank = Suggestion does not include a range.
Sugg Full Addressline	Outputs the complete address line, including firm name, secondary address, and dual address (street and postal) as appropriate for the country. The default is <i>No</i> .
Sugg Full Lastline	Outputs the locality, region, and postal code together in one component as appropriate for the country. The default is <i>No</i> .
Sugg Single Address	Outputs the combined result of the full address line and full last line in the order appropriate for the country. The default is <i>No</i> .

8.5.9.13 Global Address Cleanse fields

The Global Address Cleanse transform requires that you map fields on input and output.

[Field category columns in Output tab \[page 598\]](#)

[Input fields for the Global Address Cleanse transform \[page 601\]](#)

[NW input fields \[page 605\]](#)

This topic contains a list of NW input fields and their descriptions.

[Mapping NW input fields \[page 607\]](#)

[Output fields for the Global Address Cleanse transform \[page 608\]](#)

[NW_PO_Box output fields \[page 631\]](#)

[Global Address Cleanse Suggestion List fields \[page 633\]](#)

The Global Address Cleanse transform's Suggestion List option requires that you map fields on input and output.

Parent topic: [Global Address Cleanse \[page 552\]](#)

Related Information

[Report and Analysis \[page 554\]](#)
[Reference files \[page 558\]](#)
[Country ID options \(Global Address Cleanse\) \[page 559\]](#)
[Engines \[page 561\]](#)
[Standardization Options: Country \[page 562\]](#)
[Standardization options \[page 563\]](#)
[Canada options: Global Address Cleanse \[page 572\]](#)
[Canada options: Report options \[page 575\]](#)
[Canada engine Suggestion List options \[page 576\]](#)
[Global Address \[page 578\]](#)
[USA: Global Address Cleanse \[page 589\]](#)
[Suggestion List: Global Address Cleanse \[page 592\]](#)
[Global Address Cleanse sample configurations \[page 638\]](#)

8.5.9.13.1 Field category columns in Output tab

The Output tab lists output fields that hold the data that the transform cleanses or creates. You can choose to view the Best Practice, In Use, or All output fields by selecting the corresponding option at the top of the tab.

Best Practice: Lists all available output fields that have a field class of Best.

In Use: Lists only the output fields that you have chosen to output (listed in Schema Out).

All: Lists all output fields that are available for this transform.

i Note

For details about mapping input and output fields, see the *Designer Guide*.

The output field attributes in the following table are listed in groups based on the field category column. Each field has categories that describe the type of content that is output. The field category displays “None” when it does not apply to the field.

Field category	Description
Content Type	Identifies the type of data in the field. Setting the content type helps you map your fields when you set downstream transforms. Content types identifies the type of data in the field. Setting the content type helps you map your fields when you set downstream transforms.

Field category	Description
Field Addrclass	<p>Specifies the address class for the generated field.</p> <p><i>Delivery:</i> When used with the applicable Field Name, this value generates fields that reflect the address that is used in an attempt to assign an address.</p> <p><i>Dual:</i> When used with the applicable Field Name, this value generates fields that reflect the address that is not used in an attempt to assign an address for input records that may contain both a street and postal address on input.</p> <p><i>Official:</i> When used with the applicable Field Name, this value generates fields in the form of the data preferred by the Postal Authority.</p> <p>For example, in Winona, Minnesota USA, Broadway and 6th Street are alternate names for the same street. A letter addressed to Broadway is delivered, but the USPS prefers 6th Street.</p>
Field Category	<p><i>Component:</i> Individual address components and postal codes that are related to the processed record.</p> <p><i>Standardized:</i> Standardized input lines based on the settings in the Standardization Options group in the transform.</p> <p><i>Suggestion:</i> Suggestion list output data based on the settings in the Suggestion List Options group.</p>

Field category	Description
Field Class (Delivery:USA Regulatory Address Cleanse)	<p>Specifies the field class that you want to assign to your output fields.</p> <p><i>Best</i>: Outputs data based on various factors, such as whether an address was assigned, the Field AddrClass, and any settings that you defined in the Standardization Options group in the Options tab.</p> <div> <p>i Note</p> <p>When NCOALink is enabled and a valid move is available, Best fields contain the move-updated address data if it exists and if it matches in the U.S. National Directories. Or, the field contains the original address data if a move does not exist or if the move does not match in the U.S. National Directories.</p> </div> <p><i>Correct</i>: Outputs the complete and correct value found in the directories, and is standardized according to any settings that you defined in the Standardization Options group in the Options tab.</p> <p><i>Parsed</i>: Outputs the parsed value.</p> <p><i>Pre_LACSLink</i>: Retained address components that were replaced with LACSLink address information.</p> <p><i>Move_Updated</i>: Outputs the address components that have been updated with move-updated address data.</p> <div> <p>i Note</p> <p>The transform looks for the move-updated address information in the U.S. National Directories. When the move-updated address is not found in the U.S. National Directories, the software populates the Move Updated fields with information found in the Move Update Directories only. The Move Updated fields that are populated as a result of standardizing against the U.S. National Directories is not updated.</p> </div>
Field Class (Global Address Cleanse)	<p>Specifies the field class that you want to assign to your output fields.</p> <p><i>Best</i>: Outputs data based on various factors, such as whether an address was assigned, the Field AddrClass, and any settings that you defined in the Standardization Options group in the Options tab.</p> <p><i>Parsed</i>: Outputs the parsed value.</p>
Field Name	Specifies a field name where the data is populated based on the options that you specify within this transform.
Type	Specifies the type and default length of data the output field contains; for example, varchar, date, and time.

Parent topic: [Global Address Cleanse fields \[page 597\]](#)

Related Information

[Input fields for the Global Address Cleanse transform \[page 601\]](#)

[NW input fields \[page 605\]](#)

[Mapping NW input fields \[page 607\]](#)

[Output fields for the Global Address Cleanse transform \[page 608\]](#)

[NW_PO_Box output fields \[page 631\]](#)

[Global Address Cleanse Suggestion List fields \[page 633\]](#)

[Content types \[page 445\]](#)

8.5.9.13.2 Input fields for the Global Address Cleanse transform

The following are recognized Data Services input fields that you can use in the Global Address Cleanse transform. The table also shows that each input field is available based on the engine(s) that you enable:

- Canada (C)
- Global Address (G)
- USA (U)

See the fields listed in the transform's Input tab to view each field's properties.

Input field name (Global Address Cleanse)	Description	Engine
Data_Source_ID	<p>Specifies the name of the data source that the record comes from.</p> <p>If you choose to generate data quality statistics tables, use this field as part of your mapping strategy, along with primary key fields and the output field Row_ID. This mapping identifies the source of individual records in any of the non-summary data quality statistics tables, thus enabling you to trace problem data to the data source.</p> <p>This input field is applicable for the Data Cleanse transform, Global Address Cleanse transform, and Geocoder transform.</p>	All engines

Input field name (Global Address Cleanse)	Description	Engine
Address_Line	<p>The delivery address line, for example, "123 Main Street, Unit 4."</p> <p><i>China:</i> Address_Line may represent the following address components:</p> <ul style="list-style-type: none"> • Street and street number • Building, floor, unit • Residential community <p>For example,</p> <p>晨晖路 123 号中华大厦 12 楼 1201 室</p> <p>宝山新村 100 号 201 室</p> <p><i>Japan:</i> Address_Line may represent the following address components:</p> <ul style="list-style-type: none"> • Block (chome, kumi, Hokkaido go), sub-block (banchi, gaiku, tochi kukaku), and house number (go) parts of the Japanese address. • The building name, building floor, building room parts of the Japanese address. • The P.O. Box portion of the address, if applicable. <p><i>South Korea:</i> Address_Line may represent one of the following address components:</p> <ul style="list-style-type: none"> • Land-oriented lot number address: neighborhood or village, lot number, and additional address information. For example, 효자동 200 • Road name address: Street, house number, and additional address information. For example, 월미로 377-1 	All engines
Country	The identified country name of the address.	All engines
Data_Source_ID	Specifies the input source. This field is used in reports to identify the record.	All engines
Firm	<p>The name of a company or organization. In some countries, large firms have their own postal code. If you include a Firm field in your input, this transform may assign more specific postal codes.</p> <p><i>China:</i> China does not support Firm assignment. There is no Firm data for China. If the Firm is available on input, place it in this field.</p> <p><i>Japan:</i> All Firm data for addresses in Japan should be placed in this field.</p>	All engines
Lastline	The locality, region (when it is included in the data), and postal code on one line.	All engines

Input field name (Global Address Cleanse)	Description	Engine
Locality1	<p>The city, town, or suburb.</p> <p><i>China:</i> The Prefecture level localities. Prefectures (地区 diqu), Autonomous prefectures (自治州 zizhizhou), Prefecture-level cities (地级市 dijishi), Leagues (盟 meng), or Provincial countries (省直辖区 shengzhixiaxian).</p> <p><i>Japan:</i> The city (shi), island (shima), ward (ku), county (gun) district (machi) or village (mura).</p>	All engines
Locality2	<p>Any additional city, town, or suburb information.</p> <p><i>China:</i> County level localities, counties (县 xian), autonomous counties (自治县 zizhixian), county-level cities (县级市 xianjishi), districts (市辖区 shixiaqu), banners (旗 qi), autonomous banners (自治旗 zizhiqi), forestry areas (林区 linqu), or special districts (特区 tequ).</p> <p><i>Japan:</i> Any additional ward, district, village or sub-district (aza, bu, chiwari, sen).</p> <p><i>South Korea:</i> For land-oriented lot number address input, do not include neighborhood or village in Locality2. Instead, neighborhood or village, as well as lot number, should be mapped to Address_Line or Multiline1-12.</p> <p><i>USA:</i> The Puerto Rican urbanization.</p>	G, U
Locality3	<p>Any additional city, town, or suburb information.</p> <p><i>China:</i> Township level localities, townships (乡 xiang), ethnic townships (民族乡 minzuxiang), towns (镇 zhen), subdistricts (街道办事处 jiedaobanshichu), district public offices (区公所 qugongsuo), sumu (苏木 sumu), or ethnic sumu (民族苏木 minzusumu).</p> <p><i>Japan:</i> Any additional district, village, sub-district (aza, bu, chiwari, sen, donchi, and tori), or super block (joh).</p> <p><i>South Korea:</i> For land-oriented lot number address input, do not include neighborhood or village in Locality3. Instead, neighborhood or village, as well as lot number, should be mapped to Address_Line or Multiline1-12.</p>	G
Locality4	<p>Any additional city, town, or suburb information.</p> <p><i>China:</i> Village and neighborhood level localities, such as administrative villages (行政村), neighborhood committees (社区居民委员会), neighborhoods or communities (社区), or village committees (村民委员会).</p> <p><i>Japan:</i> Any additional district, village, sub-district (aza, bu, chiwari, sen, and tori), or super block (joh).</p>	G

Input field name (Global Address Cleanse)	Description	Engine
Multiline1-12	<p>A line that may contain any data. The type of data in this line may vary from record to record.</p> <p><i>Japan:</i> Represents the lines that may contain any data with the following restrictions. The address in total has to be in the traditional order of a Japanese address. In addition, the block (chome, kumi, Hokkaido go), sub-block (banchi, gaiku, tochi kukaku), and house number (go) should be within one line on input.</p>	All engines
NW_ fields	For a list of the NW_ input fields and their descriptions, see NW input fields [page 605] .	
Postcode	<p>The postal code.</p> <p><i>USA:</i> The five-digit ZIP Code and ZIP+4.</p>	All engines
Region1	<p>The state, province, or region.</p> <p><i>China:</i> Province-level regions, provinces (省 sheng), autonomous regions (自治区 zizhiqu), municipalities (直辖市 zhixishi), or special administrative regions (特别行政区 tebie xingzhengqu).</p> <p><i>Japan:</i> Represents the prefecture (to, do, fu, ken). A prefecture is similar to a state in the U.S.</p>	All engines
Region2	The state, province, or region.	All engines
Suggestion_Reply1-6	<p>Used to input the index number that corresponds to a specific last line suggestion, an address line suggestion, or secondary list suggestion. These fields can also be used to input a street primary range or a street secondary range.</p> <p>If you want to use one field to hold all of the replies (rather than using all six reply fields), you can use the Suggestion_Reply1 field and separate the replies with a pipe ().</p> <p>When using the Suggestion_Reply1-6 fields for SAP software for street and PO Box addresses, you can insert the following symbols to indicate whether the user has accepted changes made to the street address and when they are done with the street address:</p> <ul style="list-style-type: none"> <i>asterisk plus</i> (*+): The user accepts the changes made to the street address up to the specified point and is done with the street address. <i>asterisk minus</i> (*-): The user does not accept the changes made to the street address up to the specified point and is done with the street address. 	All engines

Parent topic: [Global Address Cleanse fields \[page 597\]](#)

Related Information

[Field category columns in Output tab \[page 598\]](#)

[NW input fields \[page 605\]](#)

[Mapping NW input fields \[page 607\]](#)

[Output fields for the Global Address Cleanse transform \[page 608\]](#)

[NW_PO_Box output fields \[page 631\]](#)

[Global Address Cleanse Suggestion List fields \[page 633\]](#)

[NW input fields \[page 605\]](#)

8.5.9.13.3 NW input fields

This topic contains a list of NW input fields and their descriptions.

The NW input fields are designed to be used in conjunction with other NW input fields in SAP software. Use the fields properly to avoid unexpected results in your data.

The following table also shows that each input field is available based on the engine(s) that you enable:

- Canada (C)
- Global Address (G)
- USA (U)

See the fields listed in the transform's Input tab to view each field's properties.

Caution

Use the NW_ fields properly to avoid unexpected results in your data. For more information, see [Mapping NW input fields \[page 607\]](#).

NW input field name (Global Address Cleanse)	Description	Engine
NW_Building	Contains the building information. If you map this input field, you must also map NW_Street.	All engines
NW_City1	Contains the locality. When you map NW input fields, this is a required field. The NW_City1 and NW_City2 input fields must be mapped in sequence.	All engines
NW_City2	Contains additional locality or district information.	All engines
NW_Country	Contains the country. When you map NW input fields, this is a required field.	All engines

NW input field name (Global Address Cleanse)	Description	Engine
NW_Firm	Contains the input firm name. Including a firm name in some addresses helps the software better assign and verify addresses. Also, a firm name is required for some postal authorities for specific discounts. For example, a firm name is required for France PO Box addresses that use the CEDEX postal code numbering system. CEDEX is designed for recipients of large volumes of mail.	All engines
NW_Floor_Num	Contains the floor number. If you map this input field, you must also map NW_Street.	All engines
NW_Home_City	Contains additional locality information.	All engines
NW_House_Num1	Contains the house number. The NW_House_Num1 and NW_House_Num2 input fields must be mapped in sequence. If you map this input field, you must also map NW_Street.	All engines
NW_House_Num2	Contains additional house number information. The NW_House_Num1 and NW_House_Num2 input fields must be mapped in sequence. If you map this input field, you must also map NW_Street.	All engines
NW_Location	Contains additional street information. If you map this input field, you must also map NW_Street.	All engines
NW_PO_Box_City	Contains the locality. If any of the NW_PO_Box input fields are mapped, then all of them must be mapped.	All engines
NW_PO_Box_Country	Contains the country. If any of the NW_PO_Box input fields are mapped, then all of them must be mapped.	All engines
NW_PO_Box_Postcode	Contains the postcode. If any of the NW_PO_Box input fields are mapped, then all of them must be mapped.	All engines
NW_PO_Box_Region	Contains the state, province, or region. If any of the NW_PO_Box input fields are mapped, then all of them must be mapped.	All engines
NW_PO_Box	Contains the PO Box number. If any of the NW_PO_Box input fields are mapped, then all of them must be mapped.	All engines
NW_Postcode	Contains the postcode. When you map NW input fields, this is a required field.	All engines
NW_Region	Contains the state, province, or region. When you map NW input fields, this is a required field.	All engines
NW_Room_Num	Contains the room number. If you map this input field, you must also map NW_Street.	All engines
NW_Str_Suppl1	Contains additional street information. The NW_Str_Suppl1-3 input fields must be mapped in sequence. If you map this input field, you must also map NW_Street.	All engines

NW input field name (Global Address Cleanse)	Description	Engine
NW_Str_Suppl2	Contains additional street information. The NW_Str_Suppl1-3 input fields must be mapped in sequence. If you map this input field, you must also map NW_Street.	All engines
NW_Str_Suppl3	Contains additional street information. The NW_Str_Suppl1-3 input fields must be mapped in sequence. If you map this input field, you must also map NW_Street.	All engines
NW_Street	Contains the primary street name. When you map NW input fields, this is a required field.	All engines

Parent topic: [Global Address Cleanse fields \[page 597\]](#)

Related Information

[Field category columns in Output tab \[page 598\]](#)

[Input fields for the Global Address Cleanse transform \[page 601\]](#)

[Mapping NW input fields \[page 607\]](#)

[Output fields for the Global Address Cleanse transform \[page 608\]](#)

[NW_PO_Box output fields \[page 631\]](#)

[Global Address Cleanse Suggestion List fields \[page 633\]](#)

[Mapping NW input fields \[page 607\]](#)

8.5.9.13.4 Mapping NW input fields

The NW input fields are designed to be used in conjunction with other NW input fields when cleansing address data in SAP software. The NW fields should be used only as a group and not individually in isolation. Use the fields properly to avoid unexpected results in your data.

You cannot map multiline or Address_Line input fields when you use the NW input fields. Although the NW input fields appear discrete, they behave and are processed as multiline fields. They are mapped internally to Multiline1-12 before normal Global Address Cleanse processing is performed. If an NW input field is not mapped, the multiline that would have been mapped to it is mapped to the next available NW input field.

The following are restrictions for using NW input fields:

- You cannot use NW input fields with non-NW input fields.
- The NW_City1, NW_Country, NW_Postcode, NW_Region, and NW_Street input fields are always required.
- The NW_City1-2 input fields must be mapped in sequence.
- The NW_House_Num1-2 input fields must be mapped in sequence.
- The NW_Str_Suppl1-3 input fields must be mapped in sequence.

- If any of the NW_PO_Box input fields are mapped, then all of them must be mapped.
- If the NW_PO_Box input fields are mapped, then a minimum of the following NW street-level fields must be mapped:
 - NW_City1
 - NW_Country
 - NW_Postcode
 - NW_Region
 - NW_Street

CJK script

For CJK script input, several NW input fields are concatenated into the last multiline field.

If a descriptor is found on NW_House_Num2, the multiline is processed as follows (+ indicates that the fields are concatenated with no space between them):

```
NW_Street+NW_House_Num1+NW_House_Num2+<Descriptor>NW_RoomNumber
+<Descriptor>NW_Floor+NW_Building
```

If no descriptor is found on NW_House_Num2, the multiline is processed as follows:

```
NW_Street+NW_House_Num1+NW_RoomNumber<Descriptor>+NW_Floor<Descriptor>
+NW_Building NW_House_Num2
```

Parent topic: [Global Address Cleanse fields \[page 597\]](#)

Related Information

[Field category columns in Output tab \[page 598\]](#)

[Input fields for the Global Address Cleanse transform \[page 601\]](#)

[NW input fields \[page 605\]](#)

[Output fields for the Global Address Cleanse transform \[page 608\]](#)

[NW_PO_Box output fields \[page 631\]](#)

[Global Address Cleanse Suggestion List fields \[page 633\]](#)

[NW input fields \[page 605\]](#)

8.5.9.13.5 Output fields for the Global Address Cleanse transform

The following are Data Services output fields that can be used for the Global Address Cleanse transform. The Field_AddrClass and Field_Class values are available in the Global Address Cleanse transform's Transform Configuration Editor on the Output Best Practices tab for each field.

The table also shows that each field is available based on the engine(s) that you enable:

- Canada (C)
- Global Address (G)
- USA (U)

Output field name (Global Address Cleanse)	Description	Engine
Additional_Info1	<p><i>Austria</i>: Includes the PAC code of the currently valid address when you choose to preserve the alias address on output and the PAC code for the master record and the alias record are different.</p> <p><i>Belgium</i>: Includes the NIS code.</p> <p><i>Canada</i>: The official 13-character abbreviation of the city name, or the full spelling if the city name is less than 13 characters (including spaces).</p> <p><i>France</i>: Includes the INSEE code.</p> <p><i>Germany</i>: Includes a portion of the German freight-code (Frachtleitcode).</p> <p><i>Liechtenstein</i>: Includes the postal service district (Botenbezirke) when it is available in the data.</p> <p><i>Poland</i>: Includes the district name (powiat).</p> <p><i>Russia</i>: Includes 11 or 15 characters depending upon assignment level (PLAINCODE).</p> <p><i>South Korea</i>: Includes administration number (25-digit).</p> <p><i>Spain</i>: Includes the INE 91 section code.</p> <p><i>Switzerland</i>: Includes the postal service district (Botenbezirke) when it is available in the data.</p>	C, G
Additional_Info2	<p><i>Austria</i>: Includes the City ID (OKZ).</p> <p><i>Canada</i>: The official 18-character abbreviation of the city name, or the full spelling if the city name is less than 18 characters (including spaces).</p> <p><i>Germany</i>: Includes the District Code.</p> <p><i>Liechtenstein</i>: Additional postcode.</p> <p><i>Poland</i>: Includes the community name (gmina).</p> <p><i>Russia</i>: Specifies the standard classification codes of the districts that are used for statistical and tax purposes. Fixed at 11 characters.</p> <p><i>Spain</i>: Includes the INE Street code.</p> <p><i>Switzerland</i>: Additional postcode.</p>	C, G

Output field name (Global Address Cleanse)	Description	Engine
Additional_Info3	<p><i>Austria</i>: Includes the Pusher-Leitcode (parcel).</p> <p><i>Germany</i>: Includes the German City ID (ALORT).</p> <p><i>Russia</i>: Includes the OKTMO, which is used for statistical and tax purposes. Length of 0, 8 or 11 characters.</p> <p><i>Spain</i>: Includes the INE Town code.</p>	G
Additional_Info4	<p><i>Austria</i>: Includes the Pusher-Leitcode (letter).</p> <p><i>Russia</i>: Includes the legal entity tax code. Fixed at 4 characters.</p> <p><i>Germany</i>: Includes the German street name ID (StrSchl).</p>	G
Additional_Info5	<p><i>Austria</i>: Includes the SKZ Street Code (7-digit).</p> <p><i>Russia</i>: Includes the individual entity tax code. Fixed at 4 characters.</p> <p><i>Germany</i>: Includes the discount code for the freight-code.</p>	G
Additional_Info6	<p><i>Austria</i>: Includes the corner-house identification (1-digit). The value for a corner house is <i>1</i>.</p> <p><i>Russia</i>: Includes the Territory IFNSUL code. Fixed at 4 characters.</p>	G
Additional_Info7	<i>Russia</i> : Includes the Territory IFNSUL code. Fixed at 4 characters.	G
Additional_Info8	Reserved for future use.	All engines
Address_Line_Remainder1-4	<p>Extraneous data found in the address line, which either cannot be identified or does not belong in a standardized address.</p> <p>USA 1-2: Complete secondary non-postal address (for example, Apt. 10, Ste 500, Box 34, Rm 7, 5th Flr).</p>	All engines
Address_Type	<p>A one-character code that represents the type of address identified:</p> <p><i>P</i>: Postal</p> <p><i>S</i>: Street</p> <p><i>X</i>: Unknown</p>	All engines
Area_Name1	An industrial area such as RIICO INDUSTRIAL AREA.	G

Output field name (Global Address Cleanse)	Description	Engine
Assignment_Info	<p>Indicates whether a record is valid, invalid, or corrected, based on the status and information codes.</p> <p><i>C</i>: Corrected</p> <p><i>I</i>: Invalid</p> <p><i>V</i>: Valid</p> <p><i>B</i>: Blank</p>	All engines
Assignment_Level	<p>The level to which this transform matched the address to the data in the reference files (directories):</p> <p><i>C</i>: Country</p> <p><i>L1</i>: Locality1</p> <p><i>L2</i>: Locality2</p> <p><i>L3</i>: Locality3</p> <p><i>L4</i>: Locality4</p> <p><i>PN</i>: Primary name</p> <p><i>PR</i>: Primary range</p> <p><i>R</i>: Region</p> <p><i>S</i>: Secondary</p> <p><i>X</i>: Unknown, or the address was unassigned</p>	All engines

Output field name (Global Address Cleanse)	Description	Engine
Assignment_Type	<p>A one- or two-character code that represents the type of address.</p> <p>Engine support varies; see each code listing for supported engines.</p> <p><i>BN</i>: Building name (Canada, Global Address)</p> <p><i>F</i>: Firm (Canada, Global Address, USA)</p> <p><i>G</i>: General delivery (Canada, Global Address, USA)</p> <p><i>H</i>: High-rise building (Canada, USA)</p> <p><i>HB</i>: House Boat (Global Address)</p> <p><i>L</i>: LOT (Global Address)</p> <p><i>M</i>: Military (Canada, USA)</p> <p><i>R</i>: Rural (Canada, USA)</p> <p><i>P</i>: Postal (Canada, Global Address, USA)</p> <p><i>PI</i>: Point of reference (Global Address)</p> <p><i>PS</i>: Packstation or Paketbox (Global Address)</p> <p><i>RP</i>: Postal Served by Route (Global Address)</p> <p><i>S</i>: Street (Canada, Global Address, USA)</p> <p><i>SR</i>: Street served by route (Canada, Global Address)</p> <p><i>U</i>: Uninhabited (Global Address)</p> <p><i>W</i>: Caravan (Global Address)</p> <p><i>X</i>: Unknown or the address was unassigned (Canada, Global Address, USA)</p>	All engines
Block_Description	Block description such as "Block."	G
Block_Full	A compound output field consisting of the Block_Description and Block_Number output fields.	G
Block_Number	Block number.	G
Building_Name1	The building name for the address, which in some countries is used in place of the primary number. For example, in the U.K. an address may be "White House, High Street," where "White House" is the building name instead of a primary number in an address such as "100 High Street."	G
Building_Name1_2	A compound output field consisting of the Building_Name1 and Building_Name2 output fields.	G
Building_Name2	The building name for the address, which in some countries is used in place of the primary number.	G

Output field name (Global Address Cleanse)	Description	Engine
Building_Primary_Addr_Delivery_Dual	A compound output field consisting of the Building_Name1, Building_Name2, Primary_Address (delivery) and Primary_Address (dual) output fields.	All engines
Building_Primary_Secondary_Addr_Delivery_Dual	A compound output field consisting of the Building_Name1, Building_Name2, Primary_Secondary_Address (delivery), and Primary_Secondary_Address (dual) output fields.	All engines
Cert_Valid	Indicates a valid certification.	All engines
Country	The ISO country code or the country name of the input record. The parsed value of this component is the country data found in the input record.	All engines
Country_Name	Fully-spelled country name in the languages specified in the Output_Country_Language option.	All engines
County_Name	Fully spelled county name. <i>USA:</i> County information is not included on mail pieces.	U
Delivery_Installation_Full	A compound output field consisting of the Delivery_Installation_Name, Delivery_Installation_Qualifier, and Delivery_Installation_Type output fields.	C, G
Delivery_Installation_Name	The delivery installation city name, which is usually the same as the city name and (if it is the same) omitted from the address line. <i>Canada:</i> If the delivery installation name is different than the locality name, the delivery installation name is output to the secondary address fields. <i>Japan:</i> Returns the post office name.	C, G
Delivery_Installation_Qualifier	Delivery Installation qualifier (for example, "Main" in "RR 2 Vancouver Stn Main").	C

Output field name (Global Address Cleanse)	Description	Engine
Delivery_Installation_Type	<p>The delivery installation type.</p> <p><i>English:</i></p> <p><i>PO</i>: Post Office</p> <p><i>RPO</i>: Retail Post Outlet</p> <p><i>STN</i>: Station</p> <p><i>LCD</i>: Letter Carrier Depot</p> <p><i>CMC</i>: Community Mail Center</p> <p><i>CDO</i>: Commercial Dealership Outlet</p> <p><i>French:</i></p> <p><i>BDP</i>: Bureau de Poste</p> <p><i>CSP</i>: Comptoir Service Postal</p> <p><i>SUCC</i>: Succursale.</p> <p><i>PDF</i>: Poste de Facteurs</p> <p><i>CPC</i>: Centre Postal Communautaire</p> <p><i>CC</i>: Concession Commerciale</p>	C
Delivery_Point	<p><i>Australia</i>: Eight-digit delivery point identifier. This is the primary component needed to generate a bar-code.</p> <p>This component is not printed on mail pieces.</p> <p><i>Austria</i>: Includes the PAC code, which is a unique identifier assigned by the Austrian postal authority.</p> <p><i>New Zealand</i>: A seven-character code that represents the delivery-point identifier.</p> <p><i>United Kingdom</i>: A two-character code that represents the delivery-point suffix.</p>	G
Engine_Name	The name of the engine that was selected to process the record.	All engines

Output field name (Global Address Cleanse)	Description	Engine
Error	<p>Specifies the error status generated as the result of looking up the current record and performing suggestion processing. Possible output values are 0- 6.</p> <p>0: No suggestion selection error.</p> <p>1: Blank suggestion selection/entry.</p> <p>2: Invalid suggestion selection.</p> <p>3: Invalid primary range.</p> <p>4: Invalid floor range.</p> <p>5: Invalid unit range.</p> <p>6: Too many possible results to generate a suggestion list. Provide more information, such as a postal code, region, or locality.</p>	All engines
Extra1-12	Any non-address data found in the address block. Available only if the input data was presented through multiline fields.	All engines
Firm	<p>The firm name for the address.</p> <p>Identification of firm name data in a multiline format may be inconsistent depending upon the level of firm data available in the postal directories for each engine. To avoid inconsistent identification of firm data, use the discrete Firm field when you process multiline data.</p> <p>Canada and USA: The firm name is taken from the postal directory if found; otherwise, it's taken from the input record. Be aware that the postal directory might contain some unusual or shortened spellings that you may or may not find suitable for printing on mail pieces. If you prefer to retain your own firm data, retrieve the parsed component.</p> <p>Global Address: If the firm name is available on input or from reference data, the Global Address engine returns the firm name.</p>	All engines
Floor_Description	<p>The level description, such as "Floor."</p> <p>Japan: The level description, such as kai.</p>	All engines
Floor_Full	A compound output field consisting of the Floor_Description, Floor_Number, and Floor_Qualifier output fields.	All engines
Floor_Number	The level number or information.	All engines
Floor_Qualifier	Additional word that precedes or follows the floor information.	All engines

Output field name (Global Address Cleanse)	Description	Engine
Full_Address	The complete address line, including secondary address, and dual address (street and postal).	All engines
Info_Code	<p>If the address is not fully assigned, displays a four-character code that describes why the address could not be assigned. If the address is fully assigned, the field is blank.</p> <p>For more information, see Information codes (Global Address Cleanse) [page 808].</p>	All engines
ISO_Country_Code_2Char	The two-character ISO code that identifies a country, for example, DE is Germany.	All engines
ISO_Country_Code_3Char	The ISO-3166 three-character code that identifies a country, for example, DEU is Germany.	All engines
ISO_Country_Code_3Digit	The three-digit ISO code that identifies a country, for example, 276 is Germany.	All engines
ISO_Script_Code	The four-character script code to use for an identified country, such as LATN or KANA.	All engines
Language	The two-character ISO language code that represents the language of the address.	All engines
Lastline	The locality (Locality1–Locality4 if available), region, and postal code together in one component. The region is only included when it is required for select countries.	All engines
Lastline_Remainder1-4	Unused lastline remainder data.	G
Locality_Code	Used in some countries to distinguish sections of a large locality. For example, in France they are called arrondissements.	G
Locality1_2_Full	A compound output field consisting of the Locality1_Full and Locality2_Full output fields.	All engines
Locality1_2_Name	A compound output field consisting of the Locality1_Name and Locality2_Name output fields.	All engines
Locality1_4_Full	A compound output field consisting of the Locality1_Full, Locality2_Full, Locality3_Full, and Locality4_Full output fields.	All engines
Locality1_4_Name	A compound output field consisting of the Locality1_Name, Locality2_Name, Locality3_Name, and Locality4_Name output fields.	All engines
Locality1_Addition	Additional locality information.	G
Locality1_Alternate	Preserves the input locality if it is recognized by the postal authority as a locality name for this address. Misspellings are corrected.	C, U

Output field name (Global Address Cleanse)	Description	Engine
Locality1_Description	Locality1 descriptor. <i>Japan</i> : Locality1 descriptor. For example, shi, shima, and so on. <i>China</i> : Locality1 descriptor. For example, 市 (Shi).	G
Locality1_Full	Includes Locality1_Name, Locality_Code, Locality1_Description, and Locality1_Qualifier. It may include Locality1_Addition, depending on the standardization option settings of Locality Name Style and Include Locality Addition.	All engines
Locality1_Name	The city, town, locality, or suburb that is either the Locality1_Alternate or Locality1_Official, depending on the standardization option setting for Assign Locality. <i>Japan</i> : The city (shi), island (shima), ward (ku), county (gun), district (machi), or village (mura).	All engines
Locality1_Official	The locality name preferred by the postal authority.	All engines
Locality1_Qualifier	Used by France for Cedex.	G
Locality2_4_Full	A compound output field consisting of the Locality2_Full, Locality3_Full, and Locality4_Full output fields.	All engines
Locality2_4_Name	A compound output field consisting of the Locality2_Name, Locality3_Name, and Locality4_Name output fields.	All engines
Locality2_Description	Description of a subdivision of Locality1.	G
Locality2_Full	Includes Locality2_Name and Locality2_Description.	G, U
Locality2_Name	Additional locality information. <i>USA</i> : Urbanization (Puerto Rican addresses only).	G, U
Locality2_Official	The locality name preferred by the postal authority.	G
Locality2_Qualifier	Locality qualifier.	G
Locality3_4_Full	A compound output field consisting of the Locality3_Full and Locality4_Full output fields.	All engines
Locality3_4_Name	A compound output field consisting of the Locality3_Name and Locality4_Name output fields.	All engines
Locality3_Description	Description of a subdivision of Locality2.	G
Locality3_Full	Includes Locality3_Name and Locality3_Description.	G
Locality3_Name	Additional locality information.	G
Locality3_Official	The locality name preferred by the postal authority.	G
Locality3_Qualifier	Locality qualifier.	G

Output field name (Global Address Cleanse)	Description	Engine
Locality4_Description	Description of a subdivision of Locality3.	G
Locality4_Full	Includes Locality4_Name and Locality4_Description.	G
Locality4_Name	Additional locality information.	G
Locality4_Official	The locality name preferred by the postal authority.	G
Locality4_Qualifier	Locality qualifier.	G
Match_Block_Number	<p>A form of Block_Number that may be used in the Match transform during the comparison process. Block descriptions are not output. Data is output in uppercase, diacritical characters and apostrophes are removed, and other punctuation and multiple spaces are replaced with a single space. The field is available only as a Best component.</p> <p>For China, Russia, South Korea, and Taiwan, non-Latin scripts are transliterated to Latin to support inter-script matching. For other CJK-script countries, the field uses normal width standardization for consistency.</p>	All engines
Match_Building_Name	<p>A form of Building_Name1 that may be used in the Match transform during the comparison process. Building descriptions are output. Data is output in uppercase, diacritical characters and apostrophes are removed, and other punctuation and multiple spaces are replaced with a single space. The field is available only as a Best component.</p> <p>For China, Russia, South Korea, and Taiwan, non-Latin scripts are transliterated to Latin to support inter-script matching. For other CJK-script countries, the field uses normal width standardization for consistency.</p>	All engines
<div>i Note</div> <div>For China, building indicators will be removed.</div>		
Match_Country	<p>A form of ISO_Country_Code_2Char that may be used in the Match transform during the comparison process. Data is output in uppercase, diacritical characters and apostrophes are removed, and other punctuation and multiple spaces are replaced with a single space. The field is available only as a Best component.</p> <p>For China, Russia, South Korea, and Taiwan, non-Latin scripts are transliterated to Latin to support inter-script matching. For other CJK-script countries, the field uses normal width standardization for consistency.</p>	All engines

Output field name (Global Address Cleanse)	Description	Engine
Match_Floor_Number	<p>A form of Floor_Number that may be used in the Match transform during the comparison process. Floor descriptions and qualifiers are not output. Data is output in uppercase, diacritical characters and apostrophes are removed, and other punctuation and multiple spaces are replaced with a single space. The field is available only as a Best component.</p> <p>For China, Russia, South Korea, and Taiwan, non-Latin scripts are transliterated to Latin to support inter-script matching. For other CJK-script countries, the field uses normal width standardization for consistency.</p>	All engines
Match_Locality	<p>A form of locality that may be used in the Match transform during the comparison process. The output is not affected by standardization settings. Locality1_Official is output when a locality or better level assignment is made; otherwise, the Locality1_Name is output. Locality codes, qualifiers, or descriptions are not output. Data is output in uppercase, diacritical characters and apostrophes are removed, and other punctuation and multiple spaces are replaced with a single space. The field is available only as a Best component.</p> <p>For China, Russia, South Korea, and Taiwan, non-Latin scripts are transliterated to Latin to support inter-script matching. For other CJK-script countries, the field uses normal width standardization for consistency.</p>	All engines

Output field name (Global Address Cleanse)	Description	Engine
Match_Locality2	<p>A form of locality that may be used in the Match transform during the comparison process. The output is not affected by standardization settings. Locality2_Official is output when a locality or better level assignment is made; otherwise, the Locality2_Name is output. Locality codes, qualifiers, or descriptions are not output. Data is output in uppercase, diacritical characters and apostrophes are removed, and other punctuation and multiple spaces are replaced with a single space. The field is available only as a Best component.</p> <p>For China, Russia, South Korea, and Taiwan, non-Latin scripts are transliterated to Latin to support inter-script matching. For other CJK-script countries, the field uses normal width standardization for consistency.</p> <div> i Note For China, Japan, and South Korea, and Taiwan, Locality2-4_Official or Locality2-4_Name are output, if present. For all other countries, only Locality2_Official or Locality2_Name is output. </div>	All engines
Match_Postcode1	<p>A form of Postcode1 that may be used in the Match transform during the comparison process. Data is output in uppercase, diacritical characters and apostrophes are removed, and other punctuation and multiple spaces are replaced with a single space. The field is available only as a Best component.</p> <p>For China, Russia, South Korea, and Taiwan, non-Latin scripts are transliterated to Latin to support inter-script matching. For other CJK-script countries, the field uses normal width standardization for consistency.</p>	All engines

Output field name (Global Address Cleanse)	Description	Engine
Match_Primary_Directional	<p>A form of Primary_Prefix1 and Primary_Postfix1 that may be used in the Match transform during the comparison process. The output is not affected by standardization settings. The abbreviated form is output, if available. If a prefix and postfix are both present, they are separated by a space. Data is output in uppercase, diacritical characters and apostrophes are removed, and other punctuation and multiple spaces are replaced with a single space. The field is available only as a Best component.</p> <p>For China, Russia, South Korea, and Taiwan, non-Latin scripts are transliterated to Latin to support inter-script matching. For other CJK-script countries, the field uses normal width standardization for consistency.</p>	All engines
Match_Primary_Name	<p>A form of Primary_Name1 that may be used in the Match transform during the comparison process. The output is not affected by standardization settings. Data is output in uppercase, diacritical characters and apostrophes are removed, and other punctuation and multiple spaces are replaced with a single space. Prefix, postfix, suffix, and type data is removed. The field is available only as a Best component.</p> <p>For China, Russia, South Korea, and Taiwan, non-Latin scripts are transliterated to Latin to support inter-script matching. For other CJK-script countries, the field uses normal width standardization for consistency.</p>	All engines

Output field name (Global Address Cleanse)	Description	Engine
Match_Primary_Name2	<p>A form of Primary_Name2 that may be used in the Match transform during the comparison process. The output is not affected by standardization settings. Data is output in uppercase, diacritical characters and apostrophes are removed, and other punctuation and multiple spaces are replaced with a single space. Prefix, postfix, suffix, and type data is removed. The field is available only as a Best component.</p> <p>For China, Russia, South Korea, and Taiwan, non-Latin scripts are transliterated to Latin to support inter-script matching. For other CJK-script countries, the field uses normal width standardization for consistency.</p> <div> i Note For Brazil, China, and Japan, Primary_Name2–4 are output, if present. For all other countries, only Primary_Name2 is output. </div>	All engines
Match_Primary_Number	<p>A form of Primary_Number that may be used in the Match transform during the comparison process. Only the Primary_Number and Primary_Number_Extra are output, not the Primary_Number_Description. Data is output in uppercase, diacritical characters and apostrophes are removed, and other punctuation and multiple spaces are replaced with a single space. The field is available only as a Best component.</p> <p>For China, Russia, South Korea, and Taiwan, non-Latin scripts are transliterated to Latin to support inter-script matching. For other CJK-script countries, the field uses normal width standardization for consistency.</p>	All engines

Output field name (Global Address Cleanse)	Description	Engine
Match_Primary_Type	<p>A form of Primary_Type1 that may be used in the Match transform during the comparison process. The output is not affected by standardization settings. The abbreviated primary type is output, if available. Data is output in uppercase, diacritical characters and apostrophes are removed, and other punctuation and multiple spaces are replaced with a single space. The field is available only as a Best component.</p> <p>For China, Russia, South Korea, and Taiwan, non-Latin scripts are transliterated to Latin to support inter-script matching. For other CJK-script countries, the field uses normal width standardization for consistency.</p> <div> i Note For Brazil, China, and Japan, Primary_Type1–4 are output, if present. For all other countries, only Primary_Type1 is output. </div>	All engines
Match_Region	<p>A form of Region1_Name that may be used in the Match transform during the comparison process. Data is output in uppercase, diacritical characters and apostrophes are removed, and other punctuation and multiple spaces are replaced with a single space. The field is available only as a Best component.</p> <p>For China, Russia, South Korea, and Taiwan, non-Latin scripts are transliterated to Latin to support inter-script matching. For other CJK-script countries, the field uses normal width standardization for consistency.</p>	All engines
Match_Stairwell_Name	<p>A form of Stairwell_Name that may be used in the Match transform during the comparison process. Stairwell descriptions are not output. Data is output in uppercase, diacritical characters and apostrophes are removed, and other punctuation and multiple spaces are replaced with a single space. The field is available only as a Best component.</p> <p>For China, Russia, South Korea, and Taiwan, non-Latin scripts are transliterated to Latin to support inter-script matching. For other CJK-script countries, the field uses normal width standardization for consistency.</p>	All engines

Output field name (Global Address Cleanse)	Description	Engine
Match_Unit_Number	<p>A form of Unit_Number that may be used in the Match transform during the comparison process. Unit descriptions and qualifiers are not output. Data is output in uppercase, diacritical characters and apostrophes are removed, and other punctuation and multiple spaces are replaced with a single space. The field is available only as a Best component.</p> <p>For China, Russia, South Korea, and Taiwan, non-Latin scripts are transliterated to Latin to support inter-script matching. For other CJK-script countries, the field uses normal width standardization for consistency.</p>	All engines
Match_Wing_Name	<p>A form of Wing_Name that may be used in the Match transform during the comparison process. Wing descriptions are not output. Data is output in uppercase, diacritical characters and apostrophes are removed, and other punctuation and multiple spaces are replaced with a single space. The field is available only as a Best component.</p> <p>For China, Russia, South Korea, and Taiwan, non-Latin scripts are transliterated to Latin to support inter-script matching. For other CJK-script countries, the field uses normal width standardization for consistency.</p>	All engines
Multiline1-12	A line that may contain any data. The type of data in this line may vary from record to record.	All engines
NW_Formatted_Postcode	The postcode in a format that SAP software requires.	All engines
NW_output fields	For a list of the NW_output fields and their descriptions, see NW_PO_Box output fields [page 631]	
NW_Postcode_In_Supported_Format	Indicates whether the NetWeaver_Formatted_Postcode output field is populated.	All engines
PMB_Full	Contains private mailbox information.	All engines
PName_Secondary_Addr	Contains the full primary name (with no associated primary number) and the full secondary address.	All engines
Point_Of_Reference1_2	A compound output field consisting of the Point_of_Reference1 and Point_of_Reference2 output fields.	All engines
Point_Of_Reference1-2	A well known place or easily visible location to help locate an address. For example, Opposite to Citibank ATM.	G
Post_Office_Name	The name or numeric representation for a post office, such as, "01" BP 1012.	G

Output field name (Global Address Cleanse)	Description	Engine
Postcode_Description	<p>A word that indicates a postal code, when available on input. For example:</p> <p><i>Brazil</i>: CEP, which stands for Código de Endereçamento Postal, and is output as CEP 52041-970.</p> <p><i>China</i>: 邮编</p> <p><i>Japan</i>: 〒</p>	G
Postcode_Full	<p><i>Australia</i>: Complete four-digit postal code.</p> <p><i>Canada</i>: Complete six-character postal code (FSA + LDU).</p> <p><i>Global Address</i>: Complete postal code.</p> <p><i>USA</i>: The full ZIP Code with a hyphen (10 characters).</p> <p><i>Japan</i>: The seven-digit postal code.</p>	All engines
Postcode_In_Valid_Format	Indicates whether the postcode is in the correct format as defined by the postal authority for that country.	All engines
Postcode_Prefix	The postcode prefix that is used by some European countries. For example, many countries use the same postal code format of four or five digits. You can prefix the numeric postal code with a country code to avoid confusion when sending mail to or from the European country. The codes used are generally based on License plate codes (D for Germany or F for France) rather than ISO codes.	G
Postcode1	<p><i>Australia</i>: Four-digit postcode.</p> <p><i>Canada</i>: First three characters (FSA) of the postal code.</p> <p><i>Global Address</i>: Postal code.</p> <p><i>Japan</i>: The first three digits of the postal code.</p> <p><i>USA</i>: Five-digit primary postal code (ZIP Code). Does not include the four-digit secondary postal code (ZIP4).</p>	All engines
Postcode2	<p>The secondary postal code.</p> <p><i>Canada</i>: The last three characters (LDU) of the postal code.</p> <p><i>Japan</i>: Contains the last four digits of the postal code.</p> <p><i>USA</i>: The four-digit ZIP Code, which on a mail piece, this code follows the primary postal code with a hyphen placed between (for example, 54601-1234).</p>	All engines

Output field name (Global Address Cleanse)	Description	Engine
Primary_Address	Primary address line, such as the street address or post office box. Does not include secondary address information such as apartment. <i>Japan:</i> The full block data.	All engines
Primary_Address_Delivery_Dual	A compound output field consisting of the Primary_Address1-4 (delivery) and Primary_Address1-4 (dual) output fields.	All engines
Primary_Delivery_Mode	The delivery mode for a street served by route type address (Rural Route).	C
Primary_Delivery_Number	The delivery number for a street served by route type address (Rural Route).	C
Primary_Name_Full1	The primary name, primary type, primary prefix, and primary postfix.	All engines
Primary_Name_Full1_2	A compound output field consisting of the Primary_Name_Full1 and Primary_Name_Full2 output fields.	All engines
Primary_Name_Full1_4	A compound output field consisting of the Primary_Name_Full1, Primary_Name_Full2, Primary_Name_Full3, and Primary_Name_Full4 output fields.	All engines
Primary_Name_Full2	The primary name2, primary type2, primary prefix2, and primary postfix2.	G
Primary_Name_Full3_4	A compound output field consisting of the Primary_Name_Full3 and Primary_Name_Full4 output fields.	All engines
Primary_Name_Full3-4	The primary name and primary type.	G
Primary_Name1	The street name description (typically a street name or box description). <i>Japan:</i> Block (chome, kumi, Hokkaido go), sub-block (banchi, gaiku, tochi kukaku). The Post office name description (yuubinnyoku or siten).	All engines
Primary_Name2	Second street and name description, typically a street name or box description. <i>Japan:</i> Additional block and sub-block information.	G
Primary_Name3	The street name, delivery mode, and so on. <i>Japan:</i> Additional block and sub-block information.	G
Primary_Name4	The street name, delivery mode, and so on. <i>Japan:</i> Additional block and sub-block information.	G
Primary_Number	The premise number, rural route number, or PO Box number. In some cases it may include a range.	All engines

Output field name (Global Address Cleanse)	Description	Engine
Primary_Number_Description	<p>A description preceding the primary number. For example, KM (Kilometer) or Blk.</p> <p><i>Japan</i>: The postal number identifier 号 (go) or house number description 号 (go).</p> <p><i>China</i>: The description after street number. For example, 号(hao).</p>	G
Primary_Number_Extra	<p>Data found near the parsed primary number, which in most cases cannot be identified or does not belong in a standardized address.</p> <p><i>Japan</i>: The postal box identifier.</p>	G
Primary_Number_Full	The primary number, primary number description, and primary number extra.	All engines
Primary_Postfix1	<p>Abbreviated or non abbreviated directional (for example, N, South, NW, SE) that follows a street name. Abbreviated or non abbreviated is based on the standardization setting for Directional Style.</p> <p><i>Japan</i>: Directional that follows block or sub-block.</p>	All engines
Primary_Postfix2	<p>Abbreviated or non abbreviated directional (for example, N, South, NW, SE) that follows a street name. Abbreviated or non abbreviated is based on the standardization setting for Directional Style.</p> <p><i>Japan</i>: Directional that follows block or sub-block.</p>	G
Primary_Prefix1	<p>Abbreviated or non abbreviated directional (N, South, NW, SE) that precedes a street name. Abbreviated or non abbreviated is based on the standardization setting for Directional Style.</p> <p><i>Japan</i>: Directional that precedes block or sub-block.</p>	G, U
Primary_Prefix2	<p>Abbreviated or non abbreviated directional (N, South, NW, SE) that precedes a street name. Abbreviated or non abbreviated is based on the standardization setting for Directional Style.</p> <p><i>Japan</i>: Directional that precedes block or sub-block.</p>	G
Primary_Secondary_Addr_Delivery_Dual	A compound output field consisting of the Primary_Secondary_Address (delivery) and Primary_Secondary_Address (dual) output fields.	All engines
Primary_Secondary_Address	The primary address and secondary address in one component.	All engines
Primary_Type1	The type of primary name (some examples are rue, strasse, street, Ave, or Pl).	All engines
Primary_Type2-4	The type of primary name (some examples are rue, strasse, street, Ave, or Pl).	G

Output field name (Global Address Cleanse)	Description	Engine
Quality_Code	<p>Displays a two-character code that provides additional information about the quality of the address. The quality of the address depends on the input data, the processing engine, country, information code, and status code (if an information code is not generated).</p> <p>For more information, see Quality codes (Global Address Cleanse) [page 815].</p>	All engines
Region1	Either the Region1_Name or Region1_Symbol based on the standardization option Region Style.	All engines
Region1_2_Full	<p>A compound output field consisting of the Region1_Full and Region2_Full output fields.</p> <p>USA: Does not include Region2_Full.</p>	All engines
Region1_2_Name	<p>A compound output field consisting of the Region1_Name and Region2_Name output fields.</p> <p>USA: Does not include Region2_Name.</p>	All engines
Region1_Code	The region code, which may be the ISO region code.	All engines
Region1_Description	The Region1 description.	G
Region1_Full	Includes Region1 and Region1_Description.	All engines
Region1_Name	The fully spelled out Region1 name.	All engines
Region1_Symbol	An abbreviation of the Region1 name.	All engines
Region2	<p>Either the Region2_Name or Region2_Symbol based on the standardization option Region Style.</p> <p>USA: Contains the county name.</p>	G, U
Region2_Code	The region code, which may be the ISO region code.	All engines
Region2_Description	The Region2 description.	G
Region2_Full	Includes Region2 and Region2_Description.	G
Region2_Name	The fully spelled out Region2 name.	G
Region2_Symbol	An abbreviation of the Region2 name.	G
Remainder_Extra_PMB_Full	A compound output field consisting of the Remainder_Full, Extra1, Extra2, and PMB_Full output fields.	All engines
Remainder_Full	Contains all remainder information, including Address_Line_Remainder1-4 and Lastline_Remainder1-4.	All engines
Room_Full	A compound output field consisting of the Unit_Description (if it contains "room" or a variant) and Unit_Number output fields.	All engines
Room_Number	The unit number for units that are variations of "room" (for example, RM, RMS, ROOM, ROOMS, RM., RMS, 号室, 室, 号).	All engines

Output field name (Global Address Cleanse)	Description	Engine
Row_ID	<p>An internal ID generated by the transform that uniquely identifies a row processed by that transform.</p> <p>Use this field to join the input and output to the applicable non-summary data quality statistics table.</p> <p>The software issues a warning if you select to generate any non-summary data quality statistics tables in the transform, and you don't include the Row_ID output field. The software also issues a warning if you map this output field and don't select to generate any non-summary data quality statistics tables in the transform.</p> <p>Applicable transforms are Data Cleanse, Global Address Cleanse, and Geocoder.</p>	All engines
Secondary_Address	The block, floor, unit, stairwell, or wing data on one line.	All engines
Secondary_Address_No_Floor_No_Room	A compound output field consisting of all Secondary_Full output fields except Floor_Full and Room_Full.	All engines
Secondary_Address_No_Floor	A compound output field consisting of all Secondary_Full output fields except Floor_Full.	All engines
Secondary_Address_No_Room	A compound output field consisting of all Secondary_Full output fields except Room_Full.	All engines
Single_Address	The full address and last line in one component.	All engines
Stairwell_Description	Entrance or stairwell identifier for a building, such as, "Entrada" 1.	G
Stairwell_Full	A compound output field consisting of the Stairwell_Description and Stairwell_Name output fields.	All engines
Stairwell_Name	The name or number of an entrance or stairwell for a building, such as, Entrada "1."	G

Output field name (Global Address Cleanse)	Description	Engine
Status	<p>Specifies the suggestion status generated as the result of looking up the current record and performing suggestion processing.</p> <p><i>A</i>: Primary address-line suggestions available.</p> <p><i>AM</i>: Follow-up primary address-line suggestions available.</p> <p><i>B</i>: Primary and secondary ranges are invalid.</p> <p><i>C</i>: Address was not found.</p> <p><i>F</i>: Floor range is invalid.</p> <p><i>L</i>: Lastline suggestions available.</p> <p><i>L1</i>: Locality1 list available (when the Enable Lastline Drilldown option is Yes; Global Address engine only).</p> <p><i>L2</i>: Locality2 list available (when the Enable Lastline Drilldown option is Yes; Global Address engine only).</p> <p><i>L3</i>: Locality3 list available (when the Enable Lastline Drilldown option is Yes; Global Address engine only).</p> <p><i>L4</i>: Locality4 list available (when the Enable Lastline Drilldown option is Yes; Global Address engine only).</p> <p><i>N</i>: No suggestions available.</p> <p><i>PC</i>: Postcode suggestions available.</p> <p><i>R</i>: Primary range is invalid.</p> <p><i>R1</i>: Region1 list available (when the Enable Lastline Drilldown option is Yes; Global Address engine only).</p> <p><i>R2</i>: Region2 list available (when the Enable Lastline Drilldown option is Yes; Global Address engine only).</p> <p><i>S</i>: Unit range is invalid.</p> <p><i>U</i>: Secondary address-line suggestions available.</p> <p><i>UM</i>: Follow up secondary address-line suggestions available.</p>	All engines
Status_Code	<p>Displays a six-character code that always starts with an S. This code explains what parts of the address changed during processing.</p> <p>For more information, see Status codes (Global Address Cleanse) [page 811].</p>	All engines
Unit_Description	<p>The unit description, such as "Apartment" or "Flat."</p> <p><i>Japan</i>: The unit description, such as gousitsu.</p>	All engines

Output field name (Global Address Cleanse)	Description	Engine
Unit_Full	A compound output field consisting of the Unit Description, Unit_Number, and Unit_Qualifier output fields.	All engines
Unit_Number	The unit number, such as 100 in "Apartment 100."	All engines
Unit_Qualifier	Additional word that precedes or follows the unit information.	G
Wing_Description	Identifies a wing within a building, such as, West "Wing."	G
Wing_Full	A compound output field consisting of the Wing_Description and Wing_Name output fields.	All engines
Wing_Name	The name or number of a wing within a building, such as "West" Wing.	G

Parent topic: [Global Address Cleanse fields \[page 597\]](#)

Related Information

[Field category columns in Output tab \[page 598\]](#)

[Input fields for the Global Address Cleanse transform \[page 601\]](#)

[NW input fields \[page 605\]](#)

[Mapping NW input fields \[page 607\]](#)

[NW_PO_Box output fields \[page 631\]](#)

[Global Address Cleanse Suggestion List fields \[page 633\]](#)

[Information codes \(Global Address Cleanse\) \[page 808\]](#)

[Quality codes \(Global Address Cleanse\) \[page 815\]](#)

[Status codes \(Global Address Cleanse\) \[page 811\]](#)

[NW_PO_Box output fields \[page 631\]](#)

8.5.9.13.6 NW_PO_Box output fields

The NW_PO_Box output fields are populated only when fields are mapped to NW input fields and are used only for the PO Box address portion of SAP business suite software.

The following is a list of the available NW_PO_Box output fields. The content of each NW_PO_Box field is identical to its corresponding output field without the prefix.

- NW_PO_Box_Assignment_Info
- NW_PO_Box_Assignment_Level
- NW_PO_Box_Assignment_Type
- NW_PO_Box_Delivery_Installation_Full

- NW_PO_Box_Delivery_Point
- NW_PO_Box_Info_Code
- NW_PO_Box_ISO_Country_Code_2Char
- NW_PO_Box_ISO_Script_Code
- NW_PO_Box_Locality1_Full
- NW_PO_Box_Match_Block_Number
- NW_PO_Box_Match_Building_Name
- NW_PO_Box_Match_Country
- NW_PO_Box_Match_Floor_Number
- NW_PO_Box_Match_Locality
- NW_PO_Box_Match_Locality2
- NW_PO_Box_Match_Postcode1
- NW_PO_Box_Match_Primary_Directional
- NW_PO_Box_Match_Primary_Name
- NW_PO_Box_Match_Primary_Name2
- NW_PO_Box_Match_Primary_Number
- NW_PO_Box_Match_Primary_Type
- NW_PO_Box_Match_Region
- NW_PO_Box_Match_Stairwell_Name
- NW_PO_Box_Match_Unit_Number
- NW_PO_Box_Match_Wing_Name
- NW_PO_Box_NW_Formatted_Postcode
- NW_PO_Box_NW_Postcode_In_Supported_Format
- NW_PO_Box_Postcode_Full
- NW_PO_Box_Postcode_In_Valid_Format
- NW_PO_Box_Primary_Address
- NW_PO_Box_Primary_Number
- NW_PO_Box_Primary_Secondary_Address
- NW_PO_Box_Region1
- NW_PO_Box_Region1_Full
- NW_PO_Box_Region2
- NW_PO_Box_Region2_Full
- NW_PO_Box_Status_Code

Parent topic: [Global Address Cleanse fields \[page 597\]](#)

Related Information

[Field category columns in Output tab \[page 598\]](#)

[Input fields for the Global Address Cleanse transform \[page 601\]](#)

[NW input fields \[page 605\]](#)

[Mapping NW input fields \[page 607\]](#)

[Output fields for the Global Address Cleanse transform \[page 608\]](#)

[Global Address Cleanse Suggestion List fields \[page 633\]](#)

[Output fields for the Global Address Cleanse transform \[page 608\]](#)

8.5.9.13.7 Global Address Cleanse Suggestion List fields

The Global Address Cleanse transform's Suggestion List option requires that you map fields on input and output.

Parent topic: [Global Address Cleanse fields \[page 597\]](#)

Related Information

[Field category columns in Output tab \[page 598\]](#)

[Input fields for the Global Address Cleanse transform \[page 601\]](#)

[NW input fields \[page 605\]](#)

[Mapping NW input fields \[page 607\]](#)

[Output fields for the Global Address Cleanse transform \[page 608\]](#)

[NW_PO_Box output fields \[page 631\]](#)

[Suggestion List input fields \[page 633\]](#)

[Suggestion List output fields \[page 634\]](#)

8.5.9.13.7.1 Suggestion List input fields

The Global Address Cleanse transform's Suggestion List option supports all Global Address Cleanse input fields in addition to the suggestion reply fields.

Input field name (Global Address Cleanse Suggestion List)

Description

Suggestion_Reply1-6	<p>Used to input the index number that corresponds to a specific last line suggestion, an address line suggestion, or secondary list suggestion. These fields can also be used to input a street primary range or a street secondary range.</p> <p>If you want to use one field to hold all of the replies (rather than using all six reply fields), you can use the Suggestion_Reply1 field and separate the replies with a pipe ().</p> <p>When using the Suggestion_Reply1-6 fields for SAP software for street and PO Box addresses, you can insert the following symbols to indicate whether the user has accepted changes made to the street address and when they are done with the street address:</p> <ul style="list-style-type: none">• <i>asterisk plus (*+)</i>: The user accepts the changes made to the street address up to the specified point and is done with the street address.• <i>asterisk minus (*-)</i>: The user does not accept the changes made to the street address up to the specified point and is done with the street address.
---------------------	--

8.5.9.13.7.2 Suggestion List output fields

Use the suggestion list output fields when you use the Global Address Cleanse suggestion list options.

The table lists the field in alphabetical order.

Output field	Description
Building_Name	The building name for the address, which in some countries is used in place of the primary number. For example, in the U.K. an address may be "White House, High Street," where "White House" is the building name instead of a primary number in an address such as "100 High Street."
Delivery_Installation_Name	The delivery installation city name, which in some cases is the same as the city name and, if it is the same, omitted from the address line.
Delivery_Installation_Qualifier	Delivery Installation qualifier, for example, "Main" in "RR 2 Vancouver Stn Main".

Output field	Description
Delivery_Installation_Type	<p>The delivery installation type.</p> <p><i>English:</i></p> <ul style="list-style-type: none"> • <i>PO</i>: Post Office. • <i>RPO</i>: Retail Post Outlet. • <i>STN</i>: Station. • <i>LCD</i>: Letter Carrier Depot. • <i>CMC</i>: Community Mail Center. • <i>CDO</i>: Commercial Dealership Outlet. <p><i>French:</i></p> <ul style="list-style-type: none"> • <i>BDP</i>: Bureau de Poste. • <i>CSP</i>: Comptoir Service Postal. • <i>SUCC</i>: Succursale. • <i>PDF</i>: Poste de Facteurs. • <i>CPC</i>: Centre Postal Communautaire. • <i>CC</i>: Concession Commerciale.
Firm	The firm name for the address.
Floor_Description	The level description, such as "Floor."
Floor_Number_High	If the floor number is a range such as 20-22, LOW contains "20" and HIGH contains "22." If the floor number is not a range, both fields contain the floor number (for example, "20" and "20").
Floor_Number_Low	
Locality1-4	The city, town or suburb and any additional related information.
Locality1-4_Official	The locality name preferred by the postal authority.

Output field	Description
More_Suggestions_Available	<p>Indicates that there are more suggestions to view:</p> <ul style="list-style-type: none"> • <i>Y</i>: There are more suggestions to view • <i>N</i>: There are no more suggestions to view <p>A <i>Y</i> output value indicates that more than one page of data is available for the current address. Use the reply fields to specify what page of data to retrieve by using the key-word <i>GOTOPAGE(N)</i>, where "N" is the page to retrieve. It is not necessary to replace "N" with a page number in the reply fields, as the first page is always the default.</p> <div> <p>❖ Example</p> <p>When <i>More_Suggestions_Available</i> = <i>Y</i>, the software enables a page up or page down mechanism to give the user access to all of the data. When the user selects page down, the address is reprocessed with a reply of GOTOPAGE(2).</p> </div> <p>The actual number of pages available depends on the following factors:</p> <ul style="list-style-type: none"> • The number of suggestions the software finds for the address. • Your settings in the transform options <i>Max_Number_Address_Lines</i> and <i>Max_Number_lastlines</i>.
Postcode_Full	<p>The postal code.</p> <p><i>USA</i>: The five-digit ZIP Code</p> <p>Does not include the four-digit secondary postal code (ZIP4).</p>
Postcode1	<p><i>Australia</i>: Four-digit postcode.</p> <p><i>Canada</i>: First three characters (FSA) of the postal code.</p> <p><i>Global</i>: Postal code.</p> <p><i>USA</i>: Five-digit primary postal code (ZIP Code).</p> <p>Does not include the four-digit secondary postal code (ZIP4).</p>
Postcode2	<p>The secondary postal code.</p> <p><i>Canada</i>: The last three characters (LDU) of the postal code.</p>
Primary_Name1	The street name description (typically a street name or box description).
Primary_Name2	Second street name and description, typically a street name or box description.
Primary_Name3	The street name, delivery mode, and so on.
Primary_Name4	
Primary_Name_Full1	The primary name, primary type, primary prefix, and primary postfix.
Primary_Name_Full2	

Output field	Description
Primary_Name_Full3	The primary name and primary type.
Primary_Name_Full4	
Primary_Number_Description	A description preceding the primary number. For example, KM (Kilometer) or Blk.
Primary_Number_Extra	Data found near the parsed primary number, which in most cases cannot be identified or does not belong in a standardized address.
Primary_Number_Full	The primary number, primary number description, and primary number extra.
Primary_Number_High	If the house number is a range such as 100-102, LOW contains "100" and HIGH contains "102." If the house number is not a range, both fields contain the house number (for example, "100" and "100").
Primary_Number_Low	
Primary_Postfix1	Abbreviated or non-abbreviated directional (for example, N, South, NW, SE) that follows a street name.
Primary_Postfix2	
Primary_Prefix1	Abbreviated or non-abbreviated directional (N, South, NW, SE) that precedes a street name.
Primary_Prefix2	
Primary_Side_Indicator	<p>Indicates if even, odd, or both values are valid. This applies to streets and PO Boxes.</p> <p><i>E</i>: The record covers the even-numbered value.</p> <p><i>O</i>: The record covers the odd-numbered value.</p> <p><i>B</i>: The record covers both the even- and odd-numbered values.</p>
Primary_Type1-4	The type of primary name (rue, strasse, street, Ave, or Pl).
Region1	Returns the state, province, or region.
Secondary_Side_Indicator	<p>Indicates if even, odd, or both values are valid. This applies to floors and units.</p> <p><i>E</i>: The secondary record covers the even-numbered value.</p> <p><i>O</i>: The secondary record covers the odd-numbered value.</p> <p><i>B</i>: The secondary record covers both the even- and odd-numbered values.</p>
Selection	A unique index number that identifies this suggestion from the others in the returned list. The suggestion "selection" number ranges from 1 to the number of suggestion selections in the suggestion list.
Stairwell_Description	Entrance or stairwell identifier for a building, such as, "Entrada" 1.
Stairwell_Name	The name or number of an entrance or stairwell for a building, such as Entrada "1."
Sugg_Full_Lastline	A compound output field consisting of the locality, region, and postal code output fields as appropriate for the country.

Output field	Description
Sugg_Full_Addressline	A compound output field consisting of the complete address line, including firm name, secondary address, and dual address (street and postal) output fields as appropriate for the country.
Sugg_Single_Address	A compound output field consisting of the full address-line and full last-line output fields in the order appropriate for the country.
Unit_Description	The unit description, such as "Apartment" or "Flat."
Unit_Number_High	If the unit number is a range such as 20-22, LOW contains "20" and HIGH contains "22." If the unit number is not a range, both fields contain the unit number (for example, "20" and "20").
Unit_Number_Low	

8.5.9.14 Global Address Cleanse sample configurations

For specialized processes like cleansing address data in Australia or Brazil, Data Services has Global Address Cleanse sample transform configurations that you can include in your data flows. Find the sample transform configurations in Data Services Object Library under Global_Address_Cleanse.

Note

Sample configurations include all required options except input fields. All sample configurations display in the designer as Global_Address_Cleanse.

Sample transform name	Description
Australia_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Australia.
Brazil_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Brazil.
Canada_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Canada.
China_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in China.
Europe_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in multiple European countries.
France_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in France.
Germany_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Germany.

Sample transform name	Description
Global_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse Latin script address data in any supported country.
GlobalSuggestions_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse Latin-1 address data in any supported country using the Suggestion List feature.
Greece_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Greece when the address data consists of Greek Data
Italy_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Italy.
Japan_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Japan when the address data consists of Japanese Kanji, Katakana, and Hiragana.
Portugal_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Portugal.
Spain_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in Spain.
UK_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in the United Kingdom.
USA_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in the United States.
USASuggestions_AddressCleanse	A sample Global Address Cleanse transform configured to cleanse address data in the United States using the Suggestion List feature.

Parent topic: [Global Address Cleanse \[page 552\]](#)

Related Information

[Report and Analysis \[page 554\]](#)

[Reference files \[page 558\]](#)

[Country ID options \(Global Address Cleanse\) \[page 559\]](#)

[Engines \[page 561\]](#)

[Standardization Options: Country \[page 562\]](#)

[Standardization options \[page 563\]](#)

[Canada options: Global Address Cleanse \[page 572\]](#)

[Canada options: Report options \[page 575\]](#)

[Canada engine Suggestion List options \[page 576\]](#)


[Global Address \[page 578\]](#)

[USA: Global Address Cleanse \[page 589\]](#)
[Suggestion List: Global Address Cleanse \[page 592\]](#)
[Global Address Cleanse fields \[page 597\]](#)

8.5.10 Global Suggestion List

The Global Suggestion List transform queries addresses with minimal data, and offers suggestions to complete the address.

Global Suggestion List information

Characteristic	Description
	Global Suggestion List icon
Use	<p>The Global Suggestion List transform is A beneficial re-search tool for managing unassigned addresses from a batch process.</p> <p>Integrate this transform into your custom applications via the Web Service. If you are a programmer looking for details about how to integrate Global Suggestion List functionality, see the <i>Integrator Guide</i>.</p> <p>The Global Suggestion List transform requires the two character ISO country code on input. Therefore, you may want to place a transform, such as the Country ID transform, that outputs the ISO_Country_Code_2Char field before the Global Suggestion List transform.</p> <p>Use the Global Suggestion List transform with the Canada, Global Address, and USA engines.</p> <div><p>i Note</p><p>Global Suggestion List does not support Chinese and Japanese addresses.</p></div> <div><p>i Note</p><p>If you use the Canada engine, USA engine, or Global Address engine for Australia and New Zealand, you cannot certify your mailing for SERP, CASS, AMAS, or New Zealand certification.</p></div>

Characteristic	Description
Content objects	Transform configurations, blueprints, and other content objects.
Global Suggestion List option groups [page 641] Data Services groups the options for the Global Suggestion List transform into categories.	
Input fields (Global Suggestion List) [page 646] Map the input fields that you want to include in the transform processes.	
Output fields (Global Suggestion List) [page 648] Map output fields based on the information that you want to output from the transform.	

Parent topic: [Data Quality transforms \[page 437\]](#)

Related Information

[Blueprints and other content objects for download \[page 438\]](#)
[About Data Quality fields \[page 445\]](#)
[About data quality statistics \[page 448\]](#)
[Associate \[page 454\]](#)
[Country ID \[page 472\]](#)
[Data Cleanse \[page 474\]](#)
[DSF2® Walk Sequencer \[page 514\]](#)
[Geocoder \[page 524\]](#)
[Global Address Cleanse \[page 552\]](#)
[Match \[page 651\]](#)
[USA Regulatory Address Cleanse \[page 710\]](#)
[Address Cleanse reference \[page 765\]](#)
[Data Cleanse reference \[page 823\]](#)
[Global Suggestion List option groups \[page 641\]](#)
[Transform configurations \[page 440\]](#)
[Blueprints and other content objects for download \[page 438\]](#)

8.5.10.1 Global Suggestion List option groups

Data Services groups the options for the Global Suggestion List transform into categories.

The Global Suggestion List transform includes the following option groups.

i Note

The Global Suggestion List transform does not support Chinese and Japanese addresses.

Option group	Description
Common: Run as Separate Process	Splits the transform into a separate process.
Engines	<p>Sets the engines for processing. The default value for each country engine is Yes. Select No to disable the country engine. Countries include:</p> <ul style="list-style-type: none"> • Canada • Global Address • USA
Canada	Sets the directory path for your address cleanse reference files.
Global Address	Sets the directory path for your address cleanse reference files.
USA	Sets the directory path for your address cleanse reference files.
Reference Files	Specifies the location of your reference files. You can use a substitution variable.
Options	Set options that define how to process data with the Global Suggestion List transform.
Suggestion List	Set options that define how the transform outputs suggestion lists as well as define the output fields where Data Services posts suggestion information.

[Engines \(Global Suggestion List\) \[page 642\]](#)

The Engines option group allows you to enable or disable individual engines of the Global Suggestion List transform.

[Options \(Global Suggestion List\) \[page 643\]](#)

This option group contains all of the settings that you need to define when you process data with the Global Suggestion List transform.

[Suggestion List \(Global Suggestion List\) \[page 644\]](#)

Use this option group to configure options for how suggestion lists are output, as well as set up the fields where the suggestion information is posted.

Parent topic: [Global Suggestion List \[page 640\]](#)

Related Information

[Input fields \(Global Suggestion List\) \[page 646\]](#)

[Output fields \(Global Suggestion List\) \[page 648\]](#)

8.5.10.1.1 Engines (Global Suggestion List)

The Engines option group allows you to enable or disable individual engines of the Global Suggestion List transform.

i Note

The Suggestion List option does not support Chinese and Japanese addresses.

Option	Description
Canada	<p>Specifies if the engine is enabled or disabled for suggestion List processing. Choose one of the following:</p> <p>Yes: Enables the engine.</p> <p>No: Disables the engine.</p>
Global Address	<p>Specifies if the engine is enabled or disabled for suggestion List processing. Choose one of the following:</p> <p>Yes: Enables the engine.</p> <p>No: Disables the engine.</p>
USA	<p>Specifies if the engine is enabled or disabled for suggestion List processing. Choose one of the following:</p> <p>Yes: Enables the engine.</p> <p>No: Disables the engine.</p>

Parent topic: [Global Suggestion List option groups \[page 641\]](#)

Related Information

[Options \(Global Suggestion List\) \[page 643\]](#)

[Suggestion List \(Global Suggestion List\) \[page 644\]](#)

8.5.10.1.2 Options (Global Suggestion List)

This option group contains all of the settings that you need to define when you process data with the Global Suggestion List transform.

i Note

The Global Suggestion List transform does not support Chinese and Japanese addresses.

Option	Description
Default Country	<p>Specifies a country to use if the input field is not defined or if the Country input field is blank.</p> <p>Enter a valid two-character country code, or enter None if you do not want to use a default country.</p>

Option	Description
Enter Firm Data	<p>Specifies whether to request firm data when the selected suggestion does not have data available.</p> <p><i>Yes:</i> Requests firm data.</p> <p><i>No:</i> Does not request firm data.</p>
Return Single Item Lists	<p>Specifies whether the transform should return suggestion lists that have only one item.</p> <p><i>Yes:</i> Returns suggestion lists that only have one item.</p> <p><i>No:</i> If a suggestion list has only one item, then the single item is automatically selected from the suggestion list and processing continues.</p>

Parent topic: [Global Suggestion List option groups \[page 641\]](#)

Related Information

[Engines \(Global Suggestion List\) \[page 642\]](#)

[Suggestion List \(Global Suggestion List\) \[page 644\]](#)

8.5.10.1.3 Suggestion List (Global Suggestion List)

Use this option group to configure options for how suggestion lists are output, as well as set up the fields where the suggestion information is posted.

Yes: Outputs the component.

No: Does not output the component.

Suggestion List options

Option/Option group	Description
Output Style	<p>Specifies the format for the output suggestion list data.</p> <p><i>Delimited:</i> Outputs the suggestion list data in a delimited text format, with the delimiters specified in the Delimiter and Field Delimiter options.</p> <p><i>XML:</i> Outputs the suggestion list data as hierarchical XML. If you integrate suggestion lists via the web service, you are likely to use this option. You can then use the XML tools you own to parse through the suggestion list data.</p>

Option/Option group	Description
Delimiter	<p>Specifies a character to use to separate each suggestion in a suggestion list. This value is considered only if the Style option is set to Delimited.</p> <p>This value can be any character or string. Common delimiters include a pipe symbol (), or a string of multiple asterisks (***).</p> <p>This value should differ from the Field Delimiter value.</p>
Field Delimiter	<p>Specifies a character to use to separate each field in a single suggestion. This value is considered only if the Style option is set to Delimited.</p> <p>Each suggestion can be made up of one or more fields. If you choose to retrieve multiple fields per suggestion, those fields are separated with the character specified here.</p> <p>This value can be any character or string. The default value is a pipe symbol (). This value should differ from the Delimiter value.</p>

Lastline Components

Option/Option group	Description
Selection	Returns a unique index number that identifies this suggestion from the others in the returned list.
Locality1-3	Returns the city, town, or suburb. Additional locality information goes in Locality2.
City Addition	Returns unofficial city information that is associated with the locality. For example, there are two German cities named Frankfurt. The larger city of Frankfurt is called Frankfurt am Main and the smaller city is called Frankfurt (Oder). Locality1 would have Frankfurt for these two records, and City Addition would list (Oder) and am Main.
Region1	Returns the state, province, territory, or region of the address.
Postcode1	Returns the postal code or five-digit ZIP Code (USA).
Primary Names Available	<p>Indicates whether or not street data is available for a locality:</p> <p>Y: Yes, there are streets.</p> <p>N: No, there are not streets.</p>

Primary Name Components

Option/Option group	Description
Selection	Returns a unique index number that identifies this suggestion from the others in the returned list.
Primary Name1 Primary Name2	Returns the street description. For example, Primary Name1 may return "Marina" and Primary Name2 may return "The Slipway."
Locality1-3	Returns the city, town, or suburb. Additional locality information goes in Locality2.

Option/Option group	Description
City Addition	Returns unofficial city information that is associated with the locality. For example, there are two German cities named Frankfurt. The larger city of Frankfurt is called Frankfurt am Main and the smaller city is called Frankfurt (Oder). Locality1 would have Frankfurt for these two records, and City Addition would list (Oder) and am Main.
Postcode1	Returns the postal code.
Address Components	
Option/Option group	Description
Selection	Returns a unique index number that identifies this suggestion from the others in the returned list.
Primary Side Indicator	<p>Indicates if even, odd, or both values are valid. This applies to Street and PO box.</p> <p><i>E</i>: The record covers the even-numbered values.</p> <p><i>O</i>: The record covers the odd-numbered values.</p> <p><i>B</i>: The record covers both the even- and odd-numbered values.</p>
Firm	Returns the name of a firm, company, or organization.
Multiline1-6	Returns individual formatted address lines. This will not include country information, as it will be output in a separate field.
Postcode1	Returns the postal code.

Parent topic: [Global Suggestion List option groups \[page 641\]](#)

Related Information

[Engines \(Global Suggestion List\) \[page 642\]](#)

[Options \(Global Suggestion List\) \[page 643\]](#)

8.5.10.2 Input fields (Global Suggestion List)

Map the input fields that you want to include in the transform processes.

The following are input fields that you can use in the input mapping for the Global Suggestion List transform. The fields are listed alphabetically.

Note

The Global Suggestion List transform uses all fields provided on input to select a match. If the fields you include do not exist in the data, the transform returns an error stating "no results." If you do not own

address-level data for a country, do not include address-level fields in the input mapping. Remove address-level fields when address-level data is available but the address list states there are no results. Removing the address-level fields in this case allows you to view locality-level data for the address.

Note

Global Suggestion List does not support Chinese and Japanese addresses.

Input field name (Global Suggestion List)

Description

Country	Specifies the country to look up in the query. This field contains the two-character country ISO code, not a country name. If your data does not contain the country code, place a transform such as the Country_ID transform before the Global Suggestion List transform to generate the country codes. If this field is blank, the transform uses the country found in the Default Country option.
Locality1	Specifies the city, town, or suburb.
Postcode	Specifies the postal code to look up.
Primary_Name1	Specifies the primary street name to look up. For example, in "255 Main St" the primary name is "Main."
Primary_Number	Specifies the primary number to look up. For example, in "255 Main St." the primary number is "255."
Reply1-5	Contains the reply when the software requires more information to complete the query. Reply1-5 can also contain the reply when you need to make a selection from a list. Possible types of generated suggestion lists are lastline, primary name, and address.
Start_Selection	Specifies the starting list number. If left blank, the default value is 1.
Script_Default	Sets the output script. This option is valid for Greece. If the input data is non-numeric, then the script type is determined by the script of the input data.

❖ Example

When input data contains only numeric data for Greece, and this option is set to 2, the transform outputs the generated suggestion lists in Greek script. If the option is set to 1, the transform outputs the generated suggestion lists in Latin script.

Parent topic: [Global Suggestion List \[page 640\]](#)

Related Information

[Global Suggestion List option groups \[page 641\]](#)

[Output fields \(Global Suggestion List\) \[page 648\]](#)

8.5.10.3 Output fields (Global Suggestion List)

Map output fields based on the information that you want to output from the transform.

The following are fields that you can use in the output mapping for the Global Suggestion List transform. The fields are listed alphabetically.

i Note

Global Suggestion List does not support Chinese and Japanese addresses.

Output field name (Global Suggestion List)	Description
Country_Name	Returns the fully spelled country name, in English.
Data_Type	Returns a single-character code that indicates what type of additional data is required to complete processing. Possible output values are: <i>N</i> : No additional data is needed. <i>A</i> : Primary address data is needed. <i>F</i> : Firm data is needed. <i>R</i> : Primary range data is needed. <i>S</i> : Secondary range data is needed.
Error	Posts the error status generated as the result of looking up the current record and performing query processing. Possible output values are: <i>0</i> : There were no query errors. <i>1</i> : There was a system error while performing the query. <i>2</i> : The suggestion selection was invalid. For example, a selection of 8 is made and there are only 5 entries.
Firm	Specifies the firm name for the address.
Multiline1-6	Returns a line that may contain any data. The type of data in this line may vary from record to record. If you want to output the postal code, you cannot use one of these Multiline output fields. Use the Postcode1 field.
Postcode1	Returns the postcode. <i>Canada and Global Address</i> : Postal code. <i>USA</i> : Five-digit primary postal code (ZIP Code). Does not include the four-digit secondary postal code (ZIP4).

Output field name (Global Suggestion List)	Description
Status	<p>Returns a code indicating query status generated as the result of processing the input record and performing Global Suggestion List processing. Possible output values are:</p> <p><i>C</i>: Querying is complete.</p> <p><i>D</i>: More data is needed.</p> <p><i>E</i>: There was an error.</p> <p><i>P</i>: A suggestion list was generated.</p>
Suggestion_Count	<p>Returns the number of individual suggestion selections generated as the result of querying the current record. A nonnegative value is output. If the input record did not generate a suggestion list, this field contains a value of 0.</p> <p>Your application developer uses this field to know how many suggestion selections must be displayed to users of your custom application.</p>
Suggestion_List	<p>Contains the list of suggestions based on the Suggestion List Option settings that you set in the Global Suggestion List transform.</p>
Suggestion_Type	<p>Returns a code indicating what type of suggestion list was generated. Possible output values are:</p> <p><i>N</i>: No suggestion list was generated.</p> <p><i>A</i>: An address suggestion list was generated.</p> <p><i>L</i>: A lastline suggestion list was generated.</p> <p><i>S</i>: A secondary suggestion list was generated.</p>
System_Error_Description	<p>Posts the current Global Suggestion List system error as a descriptive string.</p>

Output field name (Global Suggestion List)	Description
System_Error_ Number	<p>Posts the current system error as a number. The values are:</p> <p><i>0</i>: No error.</p> <p><i>1</i>: Invalid postcode.</p> <p><i>2</i>: Invalid street.</p> <p><i>3</i>: Invalid town.</p> <p><i>4</i>: More information needed.</p> <p><i>5</i>: Street information needed.</p> <p><i>6</i>: No input given.</p> <p><i>7</i>: Postcode numeric.</p> <p><i>8</i>: Town needed.</p> <p><i>9</i>: Town or postcode needed.</p> <p><i>10</i>: No street information available.</p> <p><i>11</i>: Country blank.</p> <p><i>12</i>: Invalid country.</p> <p><i>13</i>: No results.</p> <p><i>14</i>: Address needed.</p> <p><i>15</i>: Premise needed.</p> <p><i>16</i>: Firm needed.</p>
Warning	<p>Posts the warning status generated as the result of looking up the current record and performing query processing. The values are:</p> <p><i>0</i>: There were no query warnings.</p> <p><i>1</i>: An incomplete suggestion list was generated. The maximum number of elements that can be placed in a suggestion list is 200.</p> <p><i>2</i>: An invalid premise was entered.</p> <p><i>3</i>: An invalid unit was entered.</p>

Parent topic: [Global Suggestion List \[page 640\]](#)

Related Information

[Global Suggestion List option groups \[page 641\]](#)

[Input fields \(Global Suggestion List\) \[page 646\]](#)

8.5.11 Match



The Match transform is responsible for performing matching based on the business rules you define. The transform then sends matching and unique records on to the next transform in the data flow.

For best results, the data in which you are attempting to find matches should be cleansed. Therefore, you may need to include other Data Quality transforms before the Match transform.

Match concepts

This section describes the Match transform, how it fits into a data flow, and the options you can set to conform to your business rules. The Match transform is only one tool, albeit the most important one, for you to use in your matching strategy. For more information about matching concepts and other transforms you can use to achieve the results you are looking for, see the Match section of the *Designer Guide*.

Parent topic: [Data Quality transforms \[page 437\]](#)

Related Information

[Blueprints and other content objects for download \[page 438\]](#)

[About Data Quality fields \[page 445\]](#)

[About data quality statistics \[page 448\]](#)

[Associate \[page 454\]](#)

[Country ID \[page 472\]](#)

[Data Cleanse \[page 474\]](#)

[DSF2® Walk Sequencer \[page 514\]](#)

[Geocoder \[page 524\]](#)

[Global Address Cleanse \[page 552\]](#)

[Global Suggestion List \[page 640\]](#)

[USA Regulatory Address Cleanse \[page 710\]](#)

[Address Cleanse reference \[page 765\]](#)

[Data Cleanse reference \[page 823\]](#)

[Match transform tab \[page 652\]](#)

[Group forming \[page 657\]](#)

[Match level options \[page 664\]](#)

[Post-match and Post-Association processing \[page 684\]](#)

8.5.11.1 Content objects

We provide content objects to help you create and complete transforms.

Transform configurations

A transform configuration is a transform with preconfigured input fields, output fields, and options that can be used in multiple data flows. These are useful if you repeatedly use a transform with specific options and input and output fields.

When Data Services is installed, read-only transform configurations are provided for the Data Quality transforms.

You can use transform configurations in your data flows or as an example of a typical transform. After you place an instance of the transform configuration in a data flow, you can override these preset defaults. You can also create your own transform configuration, either by replicating an existing transform configuration or creating a new one.

Sample blueprints and other objects

We have created Data Quality blueprints and other content objects to help you set up Data Services jobs. We've identified a number of common scenarios that you are likely to perform with Data Services. For each scenario, we've included a blueprint that is already set up to solve the business problem in that scenario.

Related Information

[Downloading blueprints and other content objects \[page 444\]](#)

[Transform configurations \[page 440\]](#)

8.5.11.2 Match transform options

8.5.11.2.1 Match transform tab

These options control the Match transform itself. Here you control whether to generate report data, which match engine will be processing data, and so on.

Option	Description
<i>Perform matching</i>	<p>Select to add the ability to compare records and other match-related operations.</p> <p>This option is not selected only in the Base_Match transform configuration. For all other configurations, this option is selected.</p>
<i>Match set name</i>	<p>Enter a name for this match set. Each Match transform in your data flow represents a match set.</p> <p>This option is already populated with name you chose if you used the Match wizard to generated this transform.</p> <p>Be sure that this name is unique within the data flow (it does not match the name of another match set).</p> <p>This name is used in the match reports to differentiate data processed by one match set versus another.</p>
<i>Match engine</i>	<p>Specifies the match engine to use, based on the type of data you will be processing. If you use the Multinational strategy in the Match wizard, this option is set to Latin1 for all match sets.</p> <p><i>Chinese:</i> Specifies that the Match transform will be processing Chinese data in Chinese script.</p> <p><i>Japanese:</i> Specifies that the Match transform will be processing Japanese data in Japanese script.</p> <p><i>Korean:</i> Specifies that the Match transform will be processing Korean data in Korean script.</p> <p><i>Latin1:</i> Specifies that the Match transform will be processing Latin1 data. In general, this is the data used throughout the Americas, Western Europe, Oceania, and much of Africa.</p> <p><i>Other_Non_Latin1:</i> Specifies that the Match transform will be processing non-Latin1 data, other than Chinese, Japanese, Korean, and Taiwanese, such as Russian, Greek, Hebrew, Arabic, and others.</p> <p><i>Taiwanese:</i> Specifies that the Match transform will be processing Taiwanese data in Taiwanese script.</p> <p>For optimum accuracy and performance, be sure that you have filtered your multinational data to separate match transforms with the appropriate match engine selected. The Match wizard can do this for you easily, if you select the Multinational strategy.</p> <p>By default, the Latin1 match engine is used. If you attempt to process non-latin1 data with the Latin1 engine, the results are unpredictable.</p>
<i>Generate report data</i>	<p>Specifies whether to generate report data for this transform. This option is available in every transform that generates report data.</p> <p><i>Yes:</i> Generates report data for this transform.</p> <p><i>No:</i> Turns off report data generation. If you do not need to generate reports (during testing of data flows, for example), you should set this option to No. This will improve match performance.</p>
<i>Logical source field</i>	<p>Specifies the field that contains the ID for the logical source.</p>

Option	Description
<i>Physical source field</i>	Specifies the field that contains the ID for the physical source (Reader).
<i>Run as a separate process</i>	<p>Yes: Splits the transform into a separate process.</p> <p>No: Keeps the transform in same process as the rest of the data flow.</p>

8.5.11.2.2 Match transform options: Data Salvage tab

Option	Description
<i>Match set name</i>	<p>Enter a name for this match set. Each Match transform in your data flow represents a match set.</p> <p>This option is already populated with name you chose if you used the Match wizard to generated this transform.</p> <p>Be sure that this name is unique within the data flow (it does not match the name of another match set).</p> <p>This name is used in the match reports to differentiate data processed by one match set versus another.</p>
<i>Enable data salvage</i>	<p>Select to perform data salvaging.</p> <p>If two records match, data salvaging temporarily copies data from a passenger record to the driver record after comparing the two records. The copied data is data that is found in the passenger record but is missing or incomplete in the driver record. Data salvaging prevents blank matching or initials matching from matching records that you may not want to match.</p>
<i>Perform data salvage default</i>	<p>Specifies the default value that indicates whether to perform data salvage if the data record does not contain a field with this value.</p> <p>Yes: Performs the data salvage on the driver record after it matches a passenger record.</p> <p>No: Does not perform the data salvage on the driver record after it matches a passenger record.</p>
<i>Specify data salvage by field</i>	Select to control data salvaging by means of a value in a field.
<i>Perform data salvage field</i>	Specifies the field that contains the indicator for performing the data salvage operation. Use this to override the default value.
<i>Specify data salvage by source</i>	Select to control data salvaging per source.
<i>Source</i>	Select a source from the drop-down menu. You must populate the Logical Input Source window with sources to have any appear in this drop-down list.
<i>Perform data salvage</i>	Select Yes or No to signify you want data salvaging performed on a source.

8.5.11.2.3 Input Source options

Input Source options for the [Associate](#) transform and the [Match](#) transform appear in the [Association Editor](#) or the [Match Editor](#), which you open using the [Tools](#) menu.

Use the Input Sources options to define input sources for which you want to track statistics and to provide additional functionality throughout the Match or Associate transforms. Before you define your input sources, map a field that contains the value that identifies the input source.

To access these options, select the transform in the data flow and select **Tools** > [<transform_name> Editor](#). Transform name is either Match or Associate as applicable. In the Editor, select [Input Sources](#) in the left pane.

Option	Description
Value field	Choose an input field from the drop-down list that contains the value for your sources.
Source name	Enter a name for your input source.
Source value	Enter the value from the input data that identifies records belonging to this source. Sources are created by matching this case-sensitive value to that contained in the value field specified.
Source type	<p>Choose a source type from the drop-down list.</p> <p>Normal: A Normal source contains good or eligible records.</p> <p>Suppress: A suppression source contains records that would often disqualify a record from use. For example, if you're using the software to refine a mailing list, a Suppress source removes records from the mailing. Examples:</p> <ul style="list-style-type: none">• DMA Mail Preference File• American Correctional Association prisons/jails lists• No pandering or non-responder lists• Credit card or bad-check suppression lists <p>Special: A Special source is treated like a Normal source, with one exception. A Special source is not counted in when determining whether a match group is single-source or or multi-source. A Special source can contribute records, but it's not counted toward multi-source status.</p> <p>For example, some companies use a source of seed names. These are names of people who report when they receive advertising mail, so that the mailer can measure mail delivery. Appearance on the seed source should not be counted toward multi-source status.</p>
Default source name	<p>Specifies the name of a source to assign records to that do not belong to a predefined source. This name must match the name of a predefined source, so you must define input sources first to see any items in this list.</p> <p>The default source name will also be used if the Source value is blank or if the maximum number of sources is reached (maximum of 10,000).</p>

Option	Description
<i>Auto generate sources</i>	<p>Select to create sources for each unique entry in the Value field.</p> <p>This can save you time because you won't have to manually define your input sources. The name of an automatically generated source will be the same as the value in the Value field.</p> <p>As each record is processed, Match will first check to see if the record belongs to a predefined source. If it does, Match will assign that record to that source. If the record does not belong to a predefined source, then Match will check to see if the record belongs to an auto-defined source. If the record belongs to an auto-defined source, Match will use the auto-defined source. If the input source is not defined, Match will add the definition to the list of defined sources. If the maximum number of source definitions has been reached, then instead of adding a new source definition, Match will use the default source.</p> <div> <p>i Note</p> <p>When using auto generate sources, the Source ID is case sensitive but, the Type field is not case sensitive. Auto generate sources will accept Type field values of N, n, P, p, S, or s.</p> </div>
<i>Default type</i>	<p>See the description of the Source type option in this table for information about source types. This type will be used for any source that does not already have a type defined in the Type field.</p>
<i>Type field</i>	<p>Choose an input field from the drop-down list that contains the input source type.</p> <p>The type field must have a value of:</p> <p><i>N</i>: Normal</p> <p><i>P</i>: Suppress (or Purge)</p> <p><i>S</i>: Special</p> <p>If the field is not defined or if the field value is not N, n, P, p, S, or s, the default type is used.</p> <div> <p>i Note</p> <p>This option is not case sensitive.</p> </div>

8.5.11.2.4 Source Group options

Adding a Source Group operation can provide you with additional statistics in certain Match reports.

Option	Description
<i>Source groups</i>	Create your source groups here. To get started, double-click the checkbox in the first row, and type in a name for your source group.

Option	Description
<i>Sources</i>	This list contains all of the sources defined in the Input Sources operation.
<i>Undefined action</i>	<p>Specifies the action to take if an input source does not appear in a source group.</p> <p><i>Ignore:</i> The input source does not belong to any source group.</p> <p><i>Default:</i> The input source belongs to the default source group specified in the <i>Default source group</i> option.</p> <p><i>Auto:</i></p> <ul style="list-style-type: none"> • If the <i>Source group field</i> option is not defined, then the input source will belong to a source group of the same name as the input source. The source group is created if necessary. • If the <i>Source group field</i> option is defined, then the input source belongs to the source group named in the <i>Source group field</i> option. The source group is created if necessary. • If the source group field's content is blank, then that input source will not belong to a source group (equivalent of Ignore).
<i>Default source group</i>	<p>Select a default source group name from the drop-down list. You can choose from your defined source groups.</p> <p>This option is required if you chose Default as the value for the <i>Undefined action</i> option.</p>
<i>Source group field</i>	Specifies the field that contains the value for your source groups.

8.5.11.3 Group forming

Group forming allows you to group and prioritize records for better match accuracy and efficiency.

Break groups

Break groups allow you to group records based on common field values (for example, postal code). Only records that share the same break group value will be compared with one another.

Use break groups to lower the number of comparisons needed and to increase the speed of the matching process.

Candidate selection

The process of candidate selection appends records from a relational database to an existing break group for processing.

For real-time jobs, candidate selection pulls a candidate set of records based on a single record or many records.

i Note

Candidate selection works with relational databases only; it does not work with flat files.

Group prioritization

Use group prioritization to ensure that your most complete and accurate records drive the comparison process.

Related Information

[Break group options \[page 658\]](#)

[Candidate selection options \[page 660\]](#)

[Group prioritization options: Priority Order tab \[page 465\]](#)

8.5.11.3.1 Break group options

Use the break group options to group records based on common field values.

Option	Description
Split records into break groups	<p>Select this option if you want to form break groups to reduce the total number of comparisons made.</p> <p>The most common case for deselecting this option is when have a real-time job and your data comes in as one break group. This scenario also often makes use of candidate selection (selecting a limited number of records from a relational database) for optimal real-time matching.</p> <div><p>⚠ Caution</p><p>Deselect this option with caution within a batch data flow. The size of a break group may not exceed 2 GB. If you use this option in a batch data flow, also set the Maximum allowable break group size (in records) option so that the collection does not exceed the size limit. If it does exceed the limit, the data flow will abort.</p></div> <div><p>i Note</p><p>Break group size is calculated by multiplying the record length by the number of records in the break group.</p></div>

Option	Description
<i>Field</i>	<p>Choose a mapped input field name from the drop-down menu that you want to include in the break key. Click the Add Row button to add another field.</p> <p>If you require a more complex break key, you could define that field using an upstream Query transform and select the field here.</p>
<i>Start Position</i>	<p>Enter the start position of the field. Valid values for a field of n are 1 to n and -1 to -n. Negative start values signify that the start position is counted from the right.</p> <p>For example, a field with a length of 7 contains JOHNSON. A start position of 2 would mean start with "O." A start position of -4 means start with the "N" (This would also be the case if the field has a length of 20, because the negative start value starts from the actual length of the string, not of the field).</p>
<i>Length</i>	Enter the number of characters in the field you want included in the break key.
<i>Break key case sensitive</i>	<p>Specifies whether to treat the break key as case sensitive.</p> <p>Yes: Treat the break key as case sensitive.</p> <p>No: Do not treat the break key as case sensitive.</p> <p>For example, if you create a break key using the primary name (street), separate break groups would be formed with values of "Main" and "main" when you specify that the break key is case sensitive.</p>
<i>Replace NULL with empty string</i>	<p>Specifies whether to convert NULL values with an empty string in the break key.</p> <p>Yes: Convert NULL to an empty string.</p> <p>No: Do not convert to an empty string.</p>
<i>Right pad fields with blanks</i>	<p>Because the break key is used for sorting and aggregating, it is sensitive to the position in which data is placed. By right-padding the break key fields you can help ensure that break groups are formed properly.</p> <p>If the <i>Replace NULL with empty string</i> option is set to YES and this option is set to YES, then fields with NULL values will be replaced with all spaces (to the length of the field).</p> <p>Yes: Right-pad fields with blank spaces.</p> <p>No: Do not right-pad fields.</p>
<i>Input already sorted</i>	<p>Specifies that the input data has already been sorted, and you do not want it sorted again.</p> <p>For example, if you require a more complex break key, you could use a Query transform to create it, and use the ORDER BY operation to order your data.</p> <p>Yes: The transform will not re-sort the input data.</p> <p>No: The transform will sort the break keys at runtime before forming break groups.</p>

Option	Description
<i>Maximum allowable break group size (in records)</i>	<p>Specifies the maximum number of records allowed in a break group. An empty value or zero means that there is no limit on the break group size.</p> <p>With this option, you can control the amount of memory used during processing by specifying the number of records processed at one time.</p> <p>If more records make it into a single break group than specified, then the data flow throws an error and stops.</p>

8.5.11.3.2 Candidate selection options

The candidate selection option group includes the following options:

Option	Description
<i>Datastore</i>	<p>Select a valid datastore.</p> <p>This list is populated with all valid SQL and persistent cache datastores.</p> <p>If you choose a persistent cache datastore, you will not be able to enter custom SQL.</p>
<i>Cache type</i>	<p>This option can be used to improve performance, with a trade-off of more memory consumption.</p> <p><i>No_Cache</i>: Specifies that each query will be sent to the database.</p> <p><i>Pre_Load_Cache</i>: Specifies that the entire secondary table is cached to a local disk or memory.</p>
<i>Auto-generate SQL</i>	Select to have your SQL generated by the transform. This option allows you to query a simple single table. If you need to join tables or create a complex WHERE clause, you should select the <i>Create custom SQL</i> option.
<i>Table</i>	Enter a valid table name from the datastore.
<i>Use break column from database</i>	Select this option if your database already contains a column that corresponds to the break key field.
<i>Break key field</i>	Select the column from the secondary table that contains the break key field.
<i>Create custom SQL</i>	Select to create custom SQL.
<i>Launch SQL Editor</i>	Opens the SQL editor. This button is only enabled if you select the <i>Create custom SQL</i> option.
<i>Use constant source value</i>	Select to assign records to a physical source for generating appropriate statistics.
<i>Physical source value</i>	Type a value for your physical source. This value will be placed in the physical source field you select.
<i>Physical source field</i>	Select the mapped field that contains the physical source name.

Option	Description
Add DB columns to mapping table	<p>If you are using the Create custom SQL option, clicking this button will add only the database columns that appear in the SELECT statement and in the order that they appear in the SELECT statement.</p> <p>If you are using the Auto-generate SQL option, clicking this button will add ALL database columns, in the order that they appear in the table schema.</p> <div> <p>Note</p> <p>If you do not associate an input field to any of these columns in the column mapping table, they will be removed when you close the window.</p> </div>

Column mapping table

This table allows you to specify which mapped field in the data flow each database selected field is assigned to.

Column	Description
Break key	Specifies whether this field is used as part of your break key.
Field	Each cell contains a list of the mapped names from the input fields in the transform.
DB column	Each cell contains a list of the column names in your database table or the selected columns from a custom query. Match the data of a column in your database to the data of a mapped field.

8.5.11.3.3 Group prioritization options: Priority Order tab

These options appear in both the [Match Editor](#) and the [Association Editor](#) based on which transform you are working with.

In the [Post Association Processing](#) or [Post Matching Processing](#) dialog of the `<transform_name>` [Editor](#), select to add a Group Prioritization operation to control group order in post processing.

Group forming prioritization

Use the Group prioritization operation to order records within each break group. This order controls which records are used as the drivers during the comparison process.

Post-match prioritization

Add a Group prioritization operation before a Group Statistics operation to order records within a match group to control which record is flagged as the master record of each group of matching records. Add a Group prioritization operation before a Best Record operation to order records within a match group to control the destination of data that is being propagated from other records to form a best record.

Group Prioritization Editor option description

Option	Description
<i>Prioritization name</i>	Specifies the name for this Group prioritization operation. If you have multiple operations in this Match or Associate transform, be sure to make this name unique.

Priority fields

Use the Priority fields table to order your break groups based on the content of a field (for example, dollar amount or date). Use the buttons to add, remove, and order rows. Place the primary sort field at the top of the list. The rest of the fields, in the order that they are positioned, determine the sub-sort that occurs.

Option	Description
<i>Input field</i>	Choose a field to sort your records on.
<i>Field order</i>	Specifies in which order records should be sorted.

Parent topic: [Post-association processing \[page 457\]](#)

Related Information

Best record options: [Best Record tab \[page 459\]](#)

Best record options: [Destination Protection tab \[page 462\]](#)

Group Statistics Editor options [\[page 463\]](#)

Group prioritization options: [Record Completeness tab \[page 466\]](#)

Unique ID options: [Unique ID tab \[page 467\]](#)

Unique ID options: [Destination Protection tab \[page 470\]](#)

8.5.11.3.4 Group prioritization options: Record Completeness tab

Make settings to set the priority of output data based on record completeness.

Option	Description
<i>Prioritization name</i>	Specifies the name for this Group prioritization operation. If you have multiple operations in this Match transform, be sure to make this name unique.
<i>Order records based on completeness of data</i>	Select this option to apply priority and blank penalty points to records to help control the order of your records.
<i>Define only field penalties</i>	Select this option so the software assesses penalties based on blank fields.
<i>Define priority and penalty fields</i>	Select this option when you have specific fields that contain the actual integer values for priority and blank penalty.
<i>Record priority field</i>	Choose the field that contains priority values. This field must contain an integer.
<i>Apply blank penalty field</i>	Choose the field that contains the indicator (Y or N) for applying blank penalty points to a record.
<i>Define priority and penalty based on input source</i>	Select to have your record priority and blank penalty indicator (Y or N) determined by membership in a given source.
<i>Source Name</i>	Choose an input source from the dropdown list in the Source Name column. The sources listed here are defined in the Input Source operation.
<i>Priority</i>	Type a priority value (an integer) in the Priority column. Remember that the lower the priority score, the higher the priority.
<i>Apply Blank Penalty</i>	Choose Yes or No to determine whether a blank penalty is applied to a record based on membership to this source.
<i>Default record priority</i>	<p>Specifies the default value for the record priority if:</p> <ul style="list-style-type: none">• The record does not contain a field with this value.• The field is blank for a record.• A record does not belong to any of the sources specified. <p>Remember that the lower the priority score, the higher the priority.</p>
<i>Default apply blank penalty</i>	<p>Specifies the default indicator to add blank penalty points to records with blank fields. Use the indicator when a record does not have a field that carries this indicator, if that field is blank or has invalid data, or if a record does not belong to any of the sources specified.</p> <p>Yes: Each blank penalty for a record is added to the record priority to generate an adjusted record priority score. The lower the score, the higher the priority.</p> <p>No: The software does not apply a penalty when the fields are blank.</p>
<i>Input field</i>	Displays the input fields that are available to assign a blank penalty score to.
<i>Blank penalty</i>	Assign a penalty value (an integer) to apply when the specified field is blank in a record.

Parent topic: [Post-association processing \[page 457\]](#)

Related Information

Best record options: Best Record tab [page 459]

Best record options: Destination Protection tab [page 462]

Group Statistics Editor options [page 463]

Group prioritization options: Priority Order tab [page 465]

Unique ID options: Unique ID tab [page 467]

Unique ID options: Destination Protection tab [page 470]

8.5.11.4 Matching

8.5.11.4.1 Match level options

These options affect processing at the match level only.

Person Options

Option	Description
<i>Match level name</i>	Enter a name for this match level. Be sure that this name is unique within the Match transform.
<i>Weighted match score</i>	<p>Specifies the weighted match score for this level.</p> <p>When your matching method includes weighted scoring, records are considered matches when the total contribution score is greater than or equal to this value.</p>
<i>Number of names that must match</i>	<p>Specifies the number of names that must match. This option requires that you have criteria of Person1_Given_Name1, Person1_Family_Name1, and so on.</p> <p><i>One</i>: Specifies that records are a match when at least one of the names meet the criteria.</p> <p><i>All</i>: Specifies that records are a match only when all of the names meet the criteria.</p>

Option	Description									
<i>Match on hyphenated family name</i>	<p>Specifies whether a single family (last) name in one record matches a hyphenated family name in another record. For example, this option considers whether the two records shown below are matches.</p> <p>This comparison is performed only if one field has a hyphen and the other does not.</p> <table><tr><th>Given</th><th>Family</th></tr><tr><td>Laura</td><td>Smith</td></tr><tr><td>Laura</td><td>Albers-Smith</td></tr></table> <p>This option works on a criteria named Family_Name1, Family_Name2, or Family_Name3.</p> <p>Yes: The family names match as long as the single family name in one record matches one of the hyphenated family names in another record.</p> <p>No: The hyphenated family name is considered a single family name and the comparison results in a low similarity, usually not meeting your family name criteria, resulting in a no-match.</p>	Given	Family	Laura	Smith	Laura	Albers-Smith			
Given	Family									
Laura	Smith									
Laura	Albers-Smith									
<i>Compare Given Name1 to Given Name2</i>	<p>Specifies whether the given name1 (first name) of one record is compared to the given name2 (middle name of another record).</p> <p>For example, the two records shown below could be considered duplicate records if this option is set to Yes.</p> <table><tr><th>Given name1</th><th>Given name2</th><th>Family name1</th></tr><tr><td>John</td><td></td><td>Smith</td></tr><tr><td>R</td><td>John</td><td>Smith</td></tr></table> <p>To use this option, you must have criteria named Person1_Given_Name1 and Person1_Given_Name2, Person2_Given_Name1 and Person2_Given_Name2, and/or Person3_Given_Name1 and Person3_Given_Name2.</p> <p>Yes: The Given_Name1 field of one record is compared to the Given_Name2 field of another record.</p> <p>No: The Given_Name1 field of one record is not compared to the Given_Name2 field of another record.</p>	Given name1	Given name2	Family name1	John		Smith	R	John	Smith
Given name1	Given name2	Family name1								
John		Smith								
R	John	Smith								
<i>Ignore family name when female</i>	<p>Specifies whether an adjustment occurs for family names when the given name is a female. To use this option, you must have at least these three criteria: Given_Name1, Family_Name1, and Gender.</p> <p>Yes: The Family_Name1 criteria is ignored when the given name gender is a female (Gender=5). For example, Laura Smith may match Laura Albers.</p> <p>No: The gender is not used and the matching process is performed as usual.</p>									

Address Options

Option	Description																		
<i>Match on Street and RR, or on Box</i>	<p>Specifies whether to match on PO Box only or on street, rural route, and PO Box. This option affects business and household records matching on address.</p> <p>Yes: Records are considered a match if the Boxes match. If the Boxes do not match, then the address and rural route address must pass the match criteria settings.</p> <table><thead><tr><th>firm</th><th>number</th><th>street</th><th>suffix</th><th>postal number</th><th>postcode</th></tr></thead><tbody><tr><td>Acme Hardware</td><td>100</td><td>Elm</td><td>Ave</td><td>200</td><td>02961</td></tr><tr><td>Acme Hardware</td><td>123</td><td>Main</td><td>St</td><td>200</td><td>02961</td></tr></tbody></table> <p>No: All forms of the address (street, rural route, and PO Box) must match.</p>	firm	number	street	suffix	postal number	postcode	Acme Hardware	100	Elm	Ave	200	02961	Acme Hardware	123	Main	St	200	02961
firm	number	street	suffix	postal number	postcode														
Acme Hardware	100	Elm	Ave	200	02961														
Acme Hardware	123	Main	St	200	02961														
<i>Address matches blank if Firms match</i>	<p>Specifies whether to match on firm data when other address data does not match. This only affects records when one has street information and the other has PO Box information. If both records have Street information that do not match, or if both have PO Box information that do not match, the records will not be found as duplicates.</p> <table><thead><tr><th>firm</th><th>number</th><th>street</th><th>suffix</th><th>postal number</th><th>postcode</th></tr></thead><tbody><tr><td>Acme Hardware</td><td>100</td><td>Elm</td><td>Ave</td><td></td><td>02961</td></tr><tr><td>Acme Hardware</td><td></td><td></td><td></td><td>300</td><td></td></tr></tbody></table> <p>Yes: If firm data matches and neither firm field is blank, blank matching is allowed for all address components.</p> <p>No: If firm data matches, but address data in one of the records is blank, the records will not be considered a match (unless blank matching is turned on for those address components).</p>	firm	number	street	suffix	postal number	postcode	Acme Hardware	100	Elm	Ave		02961	Acme Hardware				300	
firm	number	street	suffix	postal number	postcode														
Acme Hardware	100	Elm	Ave		02961														
Acme Hardware				300															
<i>Unique on resident if RR, but no Box</i>	<p>Specifies whether to match on Rural Route when an input record's Family Name field contains a resident-type name of Current Resident, Occupant, blank, or name not defined and a rural route address, with no box number.</p> <p>Yes: Places all records with this type of name data into the same match group.</p> <p>No: Does not place records with this type of name data into the same match group.</p>																		

Option	Description
Ignore Firm if Name matches	<p>This option works with odd abbreviations or spellings of firm names. This assumes that you are matching on two Family Names.</p> <p>Yes: Indicates matching names at the same address are matches, even if the firms don't match. This lets you catch the following match, which might otherwise have been missed.</p> <p>Rita Terranova Greenco 100 Bren Rd 55343</p> <p>Rita Terranova Eco Technologies 100 Bren Rd 55343</p> <p>No: Both the firm criteria and the address criteria must meet the minimum similarity threshold in order to match.</p>

8.5.11.4.2 Match criteria table

Use the match criteria table to navigate to a particular criteria by double-clicking a row in the table.

Use the Add, Remove, and Move buttons to adjust the quantity and order of your match criteria.

8.5.11.4.2.1 Match criteria options: Criteria Fields tab

Available criteria

Choose a criteria that best reflects the data in the field you want to compare.

Category	Description
Geographic	<p>Address_Data1-5: Use for address data that is not accounted for in other address-based criteria in the Geographic category.</p> <p>You can also use this criteria for fields that you know contain address data, but you're not sure which type it contains, or you can use it for international data that has not been parsed.</p> <hr/> <p>Address_Post_Office_Box: Post Office box number.</p> <hr/> <p>Address_Primary_Name: Street name data.</p> <hr/> <p>Address_Primary_Number: Street number data.</p> <hr/> <p>Address_Primary_Postfix: Address data that comes at the end of a street name, such as a directional.</p> <hr/> <p>Address_Primary_Prefix: Address data that comes at the beginning of a street name, such as a directional.</p> <hr/> <p>Address_Primary_Type: Data that tells what type of street it is (street, boulevard, lane, and so on).</p> <hr/> <p>Address_Private_Mail_Box: A private mail box (PMB) number. These are mail boxes that are not run by a postal authority.</p>

Category	Description
	<i>Address_Rural_Route_Box</i> : Rural-route box number (number only, without "Box" prefix).
	<i>Address_Rural_Route_Number</i> : Rural route number.
	<i>Address_Secondary_Number</i> : The number of a unit, building, floor, or room.
	<i>Country</i> : Country name.
	<i>Locality</i> : City, town, locality, or suburb.
	<i>Latitude_Longitude</i> : Latitude and longitude.
	<i>Postcode1</i> : Primary postal code.
	<i>Postcode2</i> : Secondary postal code.
	<i>Region</i> : Region data, such as state or province.
<i>Firm</i>	<i>Firm</i> : Firm name.
	<i>Firm_Data1-3</i> : Use for firm data that is not accounted for in other firm-based criteria. You can also use this criteria for fields that you know contain firm data, but you're not sure which type it contains. You can also use this for international data.
	<i>Firm_Match_Std1-6</i> : Firm match standards. The data in these fields is generated by the Data Cleanse transform or other pre-Match transforms.
	<i>Firm_Location</i> : A location within a company or organization.
	<i>Firm_Location_Match_Std1-6</i> Match standards for a location within a company or organization.
<i>Person</i>	<i>Name_Data1-3</i> : Use for name data that is not accounted for in other name-based criteria. You can also use this criteria for fields that you know contain name data, but you're not sure which type.
	<i>Person1-3_Given_Name1</i> : The given name1 (first name) of the persons.
	<i>Person1_Given_Name1_Match_Std1-6</i> : Given_Name1 (first name) match standards for the first person.
	<i>Person2_Given_Name1_Match_Std1-6</i> : Given_Name1 (first name) match standards for the second person.
	<i>Person3_Given_Name1_Match_Std1-6</i> : Given _Name1 (first name) match standards for the third person.
	<i>Person1-3_Gender</i> : Gender.
	<i>Person1-3_Family_Name1</i> : Family (last) name.
	<i>Person1-3_Family_Name1_Match_Std1-6</i> : Family_Name1 match standards for the persons in your data record.
	<i>Person1-3_Family_Name2</i> : Family (last) name. Use this field when family name data is split into two fields, for example, for cultures where they store the paternal family name and the maternal family name in different fields.
	<i>Person1-3_Family_Name2_Match_Std1-6</i> : Family_Name2 match standards for the persons in your data record.
	<i>Person1-3_Maturity_Postname</i> : Maturity postname.. For example, Sr. or Jr. (one standard per person).
	<i>Person1-3_Maturity_Postname_Match_Std1-6</i> : Maturity postname match standards for the persons in your data record.

Category	Description
	<i>Person1-3_Given_Name2</i> : Given name2 (middle name).
	<i>Person1_Given_Name2_Match_Std1-6</i> : Given_Name2 (middle name) match standards for the first person.
	<i>Person2_Given_Name2_Match_Std1-6</i> : Given_Name2 (middle name) match standards for the second person.
	<i>Person3_Given_Name2_Match_Std1-6</i> : Given_Name2 (middle name) match standards for the third person.
	<i>Person1-3_Honorary_Postname</i> : Honorary postname for up to three persons indicating certification, academic degree, or affiliation. For example, CPA.
	<i>Person1-3_Honorary_Postname_Match_Std1-6</i> : Honorary postname match standards.
	<i>Person1-3_Prename</i> : Prename (for example, Mr. or Mrs.) for up to three persons.
	<i>Person1-3_Prename_Match_Std1-6</i> : Prename match standards.
	<i>Person1-3_Title</i> : Job or occupational title of each person. For example, Manager.
	<i>Person1-3_Title_Match_Std1-6</i> : Title match standards for each person.
	<i>Social_Security_Number1-3</i> : Social Security numbers for up to three people in a record.
<i>Other</i>	<i>Date1-3</i> : Date data. For example, birthdate data.
	<i>Phone</i> : Phone number.
<i>Custom</i>	Use custom fields to match data that does not qualify for any of the specifically named criteria. If you prepared fields for matching through a custom Universal Data Cleanse solution, the Data Cleanse dictionary names appear in the Custom category.

Criteria field mapping

Option	Description
<i>Criteria field</i>	The criteria field that contains the data you want to compare.

Option	Description
<i>Input field mapped name</i>	<p>Choose the mapped name for your criteria field.</p> <p>If you choose any of the following input fields, the Match transform automatically adds the appropriate match standard fields.</p> <p>The fields displayed will vary depending on the value chosen for <i>Compare data using</i> on the Options tab.</p> <ul style="list-style-type: none"> • Firm • Firm_Location • Person*_Given_Name1 • Person*_Given_Name2 • Person*_Honorary_Postname • Person*_Maturity_Postname • Person*_Prenome • Person*_Title <p>You can then choose to include or exclude any of these from mapping.</p> <div> <p>i Note</p> <p>If you enable multiple field matching, any appropriate match standard fields are removed. If you want to include them in the match process, add them in the <i>Additional fields to match</i> table in the <i>Multiple field matching</i> section.</p> </div>

8.5.11.4.2.2 Match criteria options: Options tab

Option	Description
<i>Compare data using</i>	<p>Specifies how to handle fields where more than one word commonly exists. Only those options appropriate for the chosen value appear in the Comparison Rules table.</p> <p><i>Field Similarity:</i> If you choose Field Similarity, the transform compares the entire field's data as a single string. This algorithm is more efficient and should be used in fields that typically have just one word.</p> <p><i>Geo Proximity:</i> If you choose Geo proximity, the transform compares latitude and longitude information from different records for geographic proximity to determine if they are close enough to be considered duplicates.</p> <p><i>Numeric Difference:</i> If you choose Numeric Difference, the transform compares numeric information from different records based on numerical difference to see if they are close enough to be considered duplicates.</p> <p><i>Numeric Percent Difference:</i> If you choose Numeric percent difference, the transform compares numeric information from different records based on percentage of numerical difference to see if they are close enough to be considered duplicates.</p> <p><i>Word Similarity:</i> If you choose Word Similarity, the transform first parses the data into words and then compares the words. This algorithm is less efficient than the Field algorithm, but will do a better job comparing data that typically has more than one word in it.</p> <p>Many criteria options require this option to be set to Word Similarity.</p>

Pre-comparison options

Use these options to alter the data used for the comparison process. These options do not alter the data that is output from the Match transform.

Option	Description
<i>Field compare length</i>	<p>Specifies the number of characters in the field to compare.</p> <div><p>Note</p><p>This option will be disabled (grayed out) if the input field mapped to the main criteria (not a match standard) is not of type varchar.</p></div>

Option	Description
<i>Remove punctuation</i>	<p>Specifies whether to remove punctuation from your data to help provide more accurate matches. Be aware of the following:</p> <ul style="list-style-type: none"> • This option is valid for Latin1 data only. • Match will not remove a dash from a Family_Name* field. <p><i>Yes:</i> Removes punctuation.</p> <p><i>No:</i> Keeps punctuation in your data.</p> <div> <p>i Note</p> <p>This option works on the mapped input field used in this and other criteria. If you set this option to something different than how it is set in another criteria using this same field, it will override that setting. Before setting this option, be sure that there are no other criteria using this same field.</p> </div> <div> <p>⚠ Caution</p> <p>Setting this option and the <i>Convert text to numbers</i> option to <i>Yes</i> may produce undesirable results. For example:</p> <p>Suppose you have 1 . 23 as data in a criteria field. Setting <i>Remove punctuation</i> to <i>Yes</i> would convert this number to 123. This number would then match another value of 123, or, in the case of converting text to numbers, match a value of "one hundred twenty-three".</p> </div>
<i>Convert to uppercase</i>	<p>Specifies whether to convert all data to uppercase for matching purposes only. This option is valid for Latin1 data only.</p> <p><i>Yes:</i> Converts the data to uppercase for the comparison process.</p> <p><i>No:</i> Preserves the case of the data for the comparison process, treating A and a as different characters.</p> <div> <p>i Note</p> <p>This option works on the mapped input field used in this and other criteria. If you set this option to something different than how it is set in another criteria using this same field, it will override that setting. Before setting this option, be sure that there are no other criteria using this same field.</p> </div>

Option	Description
<i>Convert diacritical characters</i>	<p>Specifies whether to include diacritical characters in the matching process. Be aware of the following:</p> <ul style="list-style-type: none"> This option is valid for all match engine options. This option works only on upper-Latin1 characters (values between 128 and 255). If you are processing Japanese data, for example, you may have some Latin1 data mixed in. In these cases you will be able to convert diacritical characters. <p>Yes: Converts diacritical characters to the closest English ASCII equivalent for matching purposes. For example, ä converts to a.</p> <p>No: Preserves diacritical characters in the matching process, treating ä and a as not identical characters.</p> <div> <p>i Note</p> <p>This option works on the mapped input field used in this and other criteria. If you set this option to something different than how it is set in another criteria using this same field, it will override that setting. Before setting this option, be sure that there are no other criteria using this same field.</p> </div>
<i>Convert text to numbers</i>	<p>Specifies whether numbers represented as text (one, two, three) should be converted to numbers. If you choose Yes, they will be in cardinal (one = 1) or ordinal (first = 1st) format.</p> <p>Yes: Converts numbers represented as text to numbers.</p> <p>No: Leaves any numerical text intact.</p> <div> <p>i Note</p> <p>This option works on the mapped input field used in this and other criteria. If you set this option to something different than how it is set in another criteria using this same field, it will override that setting. Before setting this option, be sure that there are no other criteria using this same field.</p> </div> <div> <p>⚠ Caution</p> <p>Setting this option and the <i>Remove punctuation</i> option to Yes may produce undesirable results. For example:</p> <p>Suppose you have 1 . 23 as data in a criteria field. Setting <i>Remove punctuation</i> to Yes would convert this number to 123. This number would then match another value of 123, or, in the case of converting text to numbers, match a value of "one hundred twenty-three".</p> </div>
<i>Locale</i>	<p>Specifies the locale setting for this criteria field. If nothing is specified, or you specify DEFAULT, the default system locale in the software will be used.</p> <p>Setting this option is recommended if you plan to use the <i>Convert text to numbers</i> option.</p>

Comparison rules

Option	Description						
Approx substring adjustment score	<p>Specifies what score to give to words that were not matched to the other words in the compared string. This option is commonly used to compare family names that have varying representations in which the Substring adjustment score option is too strict to compare, such as "Cruz Rodríguez" and "C. de Rodríguez".</p> <p>Enter a value from 0 (default) to 100. Enter a value of 0 to disable the option.</p>						
Abbreviation adjustment score	<p>This option controls matching whole words to abbreviations. For example, long firm names are often abbreviated by removing letters. International Health Providers might be abbreviated to Intl Health Providers.</p> <p>For example:</p> <table><tr><th>Full word</th><th>Possible abbreviations</th></tr><tr><td>Business</td><td>Bus, Bsnss, Bss</td></tr><tr><td>Database</td><td>Dat, Db, Dse</td></tr></table> <p>As shown in the examples, abbreviation means that the first letter of the shorter word matches the first letter of the longer word, and all remaining letters of the shorter word appear in the longer word in the same order as in the shorter word. The value you enter is the score given to the letters that are in the longer word but not the shorter word.</p> <ul style="list-style-type: none">• Enter a value of 0 (zero) to disable abbreviation checking.• Enter a value greater than 0 to enable this option.• Enter a value of 100 if you want abbreviations and longer words to be considered a perfect match. <div>i Note<p>For this option to work, you must set the Compare data using option to Word Similarity.</p></div>	Full word	Possible abbreviations	Business	Bus, Bsnss, Bss	Database	Dat, Db, Dse
Full word	Possible abbreviations						
Business	Bus, Bsnss, Bss						
Database	Dat, Db, Dse						
Both fields blank operation	<p>Specifies whether to use this criteria when both of the records' fields for this criteria are blank.</p> <p>Eval: The value entered in the Both fields blank score option is used as the similarity score for this criteria.</p> <p>Ignore: This criteria is ignored in the comparison process, and its contribution to the weighted score is proportionally distributed among the remaining criteria, therefore negating any impact the contribution score may have had.</p>						
Both fields blank score	<p>Specifies the similarity score if both of the fields are blank and Both fields blank operation is set to Eval.</p> <p>Enter a value from 0 to 100.</p>						

Option	Description
<i>Check for transposed letters</i>	<p>Specifies whether the match score should be adjusted for any transposed characters encountered.</p> <p>Yes: The transform deducts half as many points for transposed characters as it deducts for other non-matching characters.</p> <p>For example:</p> <p>Comparison: Smith—Simth</p> <p>Finding: Words differ by one transposition (penalty of 1 correction)</p> <p>Percentage alike: 90%</p> <p>No: The transform handles transposed characters the same way it handles any non-matching characters.</p> <p>For example:</p> <p>Comparison: Smith—Simth</p> <p>Finding: Words differ by one transposition (penalty of 1 correction)</p> <p>Percentage alike: 80%</p>
<i>Contribution to weighted score</i>	<p>Specifies the contribution value, when you use the weighted or the combination scoring method.</p> <p>If no single criteria decides a match or no-match, the contribution score is calculated by summing the products of each criteria's score by each criteria's weight.</p> <p>Type a value between 0 and 100.</p>
<i>Distance Unit</i>	<p>Specifies the type of distance unit used to calculate the distance between two Latitude, Longitude pairs. This option is only available when the Geo Proximity option is selected. Select one of the following:</p> <ul style="list-style-type: none"> • <i>Feet</i> • <i>Kilometers</i> • <i>Meters</i> • <i>Miles</i>
<i>Enable inter-script matching</i>	<p>Enable this option if you have the same data in different scripts (writing systems). For example one record has Latin1 and other has Katakana, or one has Latin and other has Cyrillic .</p>

Option	Description
<i>Ext abbreviation adjustment score</i>	<p>This option handles a variation of the <i>Abbreviation adjustment score</i> option.</p> <p>Enter a number that adjusts the similarity score for these types of abbreviations. For example:</p> <ul style="list-style-type: none"> • Enter a value of 0 (default) to disable abbreviation checking. • Enter a value greater than 0 to enable abbreviation checking. <p>Remember the following when using this option:</p> <ul style="list-style-type: none"> • The first letter of the short word must match the first letter of the first word in the multiple-word string, and the remaining letters of the short word must be found in order in the multiple-word string. • Letters that match are given a score of 100. The remaining letters are given the score that you specify. • The two scores are proportionally combined to render the overall score. <div> <p>i Note</p> <p>For this option to work, you must set the <i>Compare data using</i> option to <i>Word Similarity</i>.</p> </div>
<i>Initials adjustment score</i>	<p>Specifies whether you want initials or acronyms to match whole words. For example, the firm name International Health Providers could match IHP.</p> <p>Enter a value from 0 to 100. Enter a value of 0 (default) to disable initial checking.</p> <p>Remember the following when using this option:</p> <ul style="list-style-type: none"> • The initial must match the first letter of the word. • The letters that match are given a score of 100. The remaining letters are given the score that you specify (from 1-100). • The two scores are proportionally combined to render the overall score. <p>If there are other words in the field that are not shortened, they are scored the usual way. For example, New York Police Department may be shortened to New York PD and still match.</p> <div> <p>i Note</p> <p>For this option to work for multiple-word abbreviations (such as International Health Providers = IHP) you must set the <i>Compare data using</i> option to Word Similarity. For this option to work for single-word abbreviations (such as Maria = M) you may set the <i>Compare data using</i> option to either Word Similarity or Field Similarity.</p> </div>
<i>Match score</i>	<p>Specifies the minimum similarity score needed for the records to be considered a match based on this criteria. Type a value from 0 to 101.</p> <p>A value of 101 ensures that this criteria alone is not enough to consider two records a match and that you want to consider other criteria in the comparison process.</p> <p>For example, a value of 90 means that you consider this data to be important enough that if the data in two records is 90% similar or higher, the records are considered a match.</p>

Option	Description
<i>No match score</i>	<p>Specifies the maximum similarity score needed for the records to be considered a no-match based on this criteria. Type a value from -1 to 100.</p> <p>A value of -1 ensures that this criteria is not enough to consider two records a no-match and that you want to consider other criteria in the comparison process.</p> <p>For example, a value of 49 means that if the similarity between the data in two records is less than 50%, the records do not match.</p>
<i>Max Difference</i>	<p>Specifies the maximum difference allowed in a numeric range. Type a value from 0 - 2147483647.</p>
<i>Max Difference Score</i>	<p>Specifies what score to generate when the difference is the same as the Max Difference. Valid values for this required attribute range from 0 to 100.</p> <p>Any difference larger than the Max Difference will receive a score of 0. A difference equal to Max Difference will receive a score of Max Difference Score.</p> <p>Any difference less than Max Difference will receive a proportional score between Max Difference Score and 100.</p>
<i>Max Distance</i>	<p>Specifies the maximum distance allowed when calculating the distance between two Latitude, Longitude pairs. Type a value from 0 to 4294967295.0.</p>
<i>Max Distance Score</i>	<p>Specifies what score to generate when the distance is the same as Max Distance. Type a value from 0 to 100.</p> <p>Any distance larger than the Max Distance will receive a score of 0.</p> <p>A distance equal to Max Distance will receive a score of Max Distance Score.</p> <p>Any distance less than Max Distance will receive a proportional score between Max Distance Score and 100.</p>
<i>Max Percent Difference</i>	<p>Specifies the maximum difference allowed as a percent of the absolute value. Type a value from 0 to 100.</p>
<i>Max Percent Difference Score</i>	<p>Specifies what score to generate when the difference is the same as Max Percent Difference. Valid values for this required attribute range from 0 to 100.</p> <p>Any difference larger than the Max Percent Difference will receive a score of 0.</p> <p>A difference equal to Max Percent Difference will receive a score of Max Percent Difference Score.</p> <p>Any difference less than Max Percent Difference will receive a proportional score between Max Percent Difference Score and 100.</p>

Option	Description
<i>Numeric words match exactly</i>	<p>Specifies how to match data that contains both numbers and letters.</p> <p><i>None</i>: Numeric words don't need to match exactly to be considered a match.</p> <p><i>Any_Position</i>: Numeric words don't need to be in the same position in two different strings to be considered a match.</p> <p><i>Any_Position_Consider_Punctuation</i>: This value behaves the same as the <i>Any_Position</i> value. However, the Match transform takes the position of a decimal separator (comma or period) within the numeric words into consideration.</p> <p><i>Any_Position_Ignore_Punctuation</i>: Same as <i>Any_Position</i>, except that decimal separators (comma or period) are completely ignored.</p> <p><i>Same_Position</i>: Numeric words must match exactly and be in the same position in the string to be considered a match.</p> <div> <p>Note</p> <p>For this option to work, you must set the <i>Compare data using</i> option to <i>Word Similarity</i>.</p> </div>
<i>One field blank operation</i>	<p>Specifies whether to use this criteria if one record's field is populated and the other record's field is blank.</p> <p><i>Eval</i>: The value entered in the One field blank score is used as the similarity score for this criteria.</p> <p><i>Ignore</i>: This criteria is ignored in the comparison process, and its contribution to the weighted score is proportionally distributed among the remaining criteria.</p>
<i>One field blank score</i>	<p>Specifies the similarity score to use if one of the fields is blank and the <i>One field blank operation</i> option is set to <i>Eval</i>.</p> <p>Type a value from 0 to 100.</p>

Option	Description
<i>Substring adjustment score</i>	<p>Allows matching longer strings of words to shorter strings. For example, long firm names are often shortened to the first few words of the name. A fictitious company such as Mayfield Painting and Sand Blasting might be shortened to Mayfield Painting.</p> <p>Remember the following rules about values to enter:</p> <ul style="list-style-type: none"> • Enter a value from 0 to 100. Enter a value of 0 (default) to disable substring checking. • Enter a value of 100 if you want substrings and longer strings to be considered a match. <p>Here is what happens after processing.</p> <ul style="list-style-type: none"> • Letters that match are given a score of 100. The remaining letters are given the score you specify (from 1-100). • The two scores are proportionally combined to render the overall score. <p>To qualify as a substring match, the shorter string must exactly match the first part of the longer string.</p> <p>Consider the following example:</p> <p>Matching substrings</p> <ul style="list-style-type: none"> • Mayfield • Mayfield Painting • Mayfield Painting and Sand <p>Substrings that do not match</p> <ul style="list-style-type: none"> • Mayfield Sand Blasting • Painting and Sand Blasting <p>Alternate spellings in any of the words also disqualify the substrings as a match. For example, "Murphy Painting and Sand Blasting" does not match. (This comparison would have a similarity score of 85% without this option set.)</p> <div> <p>i Note</p> <p>For this option to work, you must set the <i>Compare data using</i> option to <i>Word Similarity</i>.</p> </div>
<i>Use in weighted score if greater than</i>	<p>Specifies the minimum similarity score needed to qualify this criteria to contribute to the Weighted Match Score.</p> <p>For example, if the value entered here is 59 for a given name criteria, and the given names between two records are less than 60% similar, then the given name criteria is ignored and the contribution value is proportionally distributed among the remaining criteria.</p>
<i>Zero weighted score if less or equal</i>	<p>Specifies the minimum similarity score needed for this criteria to qualify for contributing a value other than zero to the weighted match score.</p> <p>For example, if the value is 59 for the given name criteria, and the given names between two records are less than 60% similar, then the given name criteria contributes zero toward the weighted match score.</p>

Reset Rules buttons

We provide you with default matching scores for every criteria (including options such as Match score, No match score, Contribution to weighted score, One field blank score, Both fields blank score, and so on). The Reset Rules buttons allow you to try different base levels of matching scores, as well as provide you a way to return to the default scores. These buttons primarily adjust the No match score and other options that are dependent on that score. The [Reset All to Default](#) button returns all options and scores to the original default values.

Loose matches mean that there will be more matches in a match group, but you may sacrifice in the quality of the matches.

Exact matches mean that the quality of the matches will be high, but they will be fewer in number.

8.5.11.4.2.3 Match criteria options: Multiple Field Comparisons tab

Multiple field mapping

Option	Description
Compare multiple fields	Enable multiple field matching.
All selected fields in other records	Match each field against all fields selected in the table in each record.
The same field in other records	Match each field only against the same field in each record.

Additional fields to compare

Option	Description
Criteria field	Choose the input field that contains the data you want to compare.
Custom name	Enter a name for your custom field.
Input field mapped name	Choose the mapped input field name for the criteria field. The fields displayed will vary depending on the value chosen for Compare data using on the Options tab.

Pre-comparison options

Use these options to optimize your data for faster comparison. These options do not alter your source data; they allow Match to change your data internally.

Option	Description
<i>Field compare length</i>	Specifies the number of characters in the field to compare.
<i>Remove punctuation</i>	<p>Specifies whether to remove punctuation from your data to help provide more accurate matches. Be aware of the following:</p> <ul style="list-style-type: none"> • This option is valid for Latin1 data only. • Match will not remove a dash from a Family_Name* field. <p>Yes: Removes punctuation.</p> <p>No: Keeps punctuation in your data.</p> <div> <p>Note</p> <p>This option works on the mapped input field used in this and other criteria. If you set this option to something different than how it is set in another criteria using this same field, it will override that setting. Before setting this option, be sure that there are no other criteria using this same field.</p> </div> <div> <p>Caution</p> <p>Setting this option and the <i>Convert text to numbers</i> option to Yes may produce undesirable results. For example:</p> <p>Suppose you have 1 . 23 as data in a criteria field. Setting <i>Remove punctuation</i> to Yes would convert this number to 123. This number would then match another value of 123, or, in the case of converting text to numbers, match a value of "one hundred twenty-three".</p> </div>
<i>Convert to uppercase</i>	<p>Specifies whether to convert all data to uppercase for matching purposes only. Be aware of the following:</p> <ul style="list-style-type: none"> • This option is valid for English language, Latin1 data only. • This option is ignored for all other Match engine option values. <p>Yes: Converts the output data to uppercase where appropriate.</p> <p>No: Leaves the output data intact.</p> <div> <p>Note</p> <p>This option works on the mapped input field used in this and other criteria. If you set this option to something different than how it is set in another criteria using this same field, it will override that setting. Before setting this option, be sure that there are no other criteria using this same field.</p> </div>

Option	Description
<i>Convert diacritical characters</i>	<p>Specifies whether to include diacritical characters in the matching process. Be aware of the following:</p> <ul style="list-style-type: none"> • This option is valid for all match engine options. • This option works best when using the Latin1 engine. If you are processing Japanese data, for example, you may have some Latin1 data mixed in. In these cases you will be able to convert diacritical characters. <p>Yes: Converts diacritical characters to the closest English ASCII equivalent for matching purposes. For example, ä converts to a.</p> <p>No: Preserves diacritical characters in the matching process, treating ä and a as not identical characters.</p> <div> <p>i Note</p> <p>This option works on the mapped input field used in this and other criteria. If you set this option to something different than how it is set in another criteria using this same field, it will override that setting. Before setting this option, be sure that there are no other criteria using this same field.</p> </div>
<i>Convert text to numbers</i>	<p>Specifies whether numbers represented as text (one, two, three) should be converted to numbers. If you choose Yes, they will be in cardinal (one = 1) or ordinal (first = 1st) format.</p> <p>Yes: Converts numbers represented as text to numbers.</p> <p>No: Leaves any numerical text intact.</p> <div> <p>i Note</p> <p>This option works on the mapped input field used in this and other criteria. If you set this option to something different than how it is set in another criteria using this same field, it will override that setting. Before setting this option, be sure that there are no other criteria using this same field.</p> </div> <div> <p>⚠ Caution</p> <p>Setting this option and the <i>Remove punctuation</i> option to Yes may produce undesirable results. For example:</p> <p>Suppose you have 1 . 23 as data in a criteria field. Setting <i>Remove punctuation</i> to Yes would convert this number to 123. This number would then match another value of 123, or, in the case of converting text to numbers, match a value of "one hundred twenty-three".</p> </div>
<i>Locale</i>	<p>Specifies the locale setting for this criteria field. If nothing is specified, or you specify DEFAULT, the default system locale in SAP Data Services will be used.</p> <p>Setting this option is recommended if you plan to use the <i>Convert text to numbers</i> option.</p>

8.5.11.4.3 Compare table

The compare table options create a table that is used to determine which record pairs qualify to be compared or which sources should be compared. If you do not include a Compare table to the Match transform, a driver record is compared with all remaining passenger records in the break group.

→ Tip

If you are using many physical or logical sources in your project, it may be easier to specify what not to compare, as opposed to what to compare. For example, say you have 10 sources: A through J. You want to compare all but A and B. Set the Default action option to Compare. Then set up a table row for both source A and source B, and set the Action options for those sources to No_Match.

Option	Description
<i>Default action</i>	<p>Specifies the action assigned to each cell of the compare table initially before any Compare actions table rows are applied.</p> <p><i>Compare</i>: Compares all of the logical sources specified in the compare table. You can then specify a pair to not be compared in the Compare actions table.</p> <p><i>No_Match</i>: Does not compare all logical sources in the compare table. You can then specify a pair to be compared in the Compare actions table.</p>
<i>Default source</i>	<p>Specifies the source that a record belongs to if that record has no field to identify it. If there are no pre-defined sources, you can type a name.</p>
<i>Define compare actions using field values</i>	<p>Select to use values in a field as the criteria for comparison, rather than membership in an input source. To use this option, you will need to enter field values in the <i>Compare Actions</i> table.</p> <p>This option is automatically selected if you do not have any input sources defined.</p>
<i>Logical source field</i>	<p>Specifies the field that contains the logical source value (name).</p>
<i>Default logical source value</i>	<p>If you are not passing in a field from another location that contains the logical source value, or there is no value in the field, then you must specify the default value here.</p> <p>This option specifies which value defined in the Compare actions table is the default value.</p>

Compare actions

Use this table to set the action of driver records and the records they are compared with (passenger records). These entries can override your Default action option value.

Option	Description
<i>Driver source</i>	<p>Specifies the source contained in the driver record. If it is omitted, then all drivers records are assumed to be the first in a break group.</p>

Option	Description
<i>Passenger source</i>	Specifies the value contained in the passenger record. If it is omitted, then all passenger records are assumed to be any record not the first in the break group.
<i>Action</i>	<p>Specifies the action to take when a record from the Driver source is to be compared with a record from the Passenger source.</p> <p><i>Compare</i>: Compare the two records</p> <p><i>No_Match</i>: Do not compare the two records.</p> <p>This option overrides the Default action option. For example, if you set the Default action option to No_Match, you can specify that you want this pair to be compared by setting this option to Compare.</p>

8.5.11.4.4 Post-match and Post-Association processing

Both the *Match Editor* and the *Associate Editor* contain a post processing page, which you use to set up any post processing processes.

Access the Post match or the Post associate options by selecting the transform icon in the data flow and selecting **Tools** > *<transform_name>Editor* > where *<transform_name>* is either Match or Associate.

Use the Post match operations or the Post associate operations table to navigate to your operations by double-clicking the desired row in the table.

Add rows to the table by clicking *Add Operation*. The following options appear in the dropdown list:

- Group Prioritization
- Group Statistics
- Best Record
- Unique ID

Group Prioritization

Use the prioritization operation to order records for processing by other post-match operations.

Group statistics

Use group statistics to generate statistical information about your group of matching records. Find out:

- The number of records within the match group.
- The sequential group order number.
- The group rank, which flags one record within each group of matching records as the Master record. Group rank groups all other records in the group as Subordinate records.

- Whether the records in a match group belong to more than one source.

Group statistics are essential for generating data for match reports.

Best record

The purpose of best record post-processing is to salvage data from matching records—that is, members of match groups—and consolidate, or post that data to a best record, or to all matching records.

Unique ID

Use the Unique ID options to assign sequential identification numbers to each new record when adding records to a data warehouse. For example, the largest number assigned in a particular project can be carried over as the beginning identification number plus 1. The new number is used in the assignment of new sequential IDs. Sequential identification numbering occurs when the software processes the next source against the data warehouse file.

Related Information

Best record options: [Best Record tab \[page 685\]](#)

Group Statistics Editor options [\[page 463\]](#)

Group forming [\[page 657\]](#)

Output flag selection options [\[page 696\]](#)

8.5.11.4.4.1 Best record options: Best Record tab

Use the best record post-match processing operation to update your records with information from other records in a match group, among other things.

These options appear in both the [Match Editor](#) and the [Association Editor](#) based on which transform you are working with.

In the [Post Association Processing](#) or [Post Matching Processing](#) dialog of the [<transform_name> Editor](#), select to add a Best Record operation. The following options are in the [Best Record Editor](#) page and the [Best Record](#) tab.

Best Record post processing operation option descriptions

Option	Description
Best record name	Enter a name for your Best Record operation. Make sure that this name is unique within this Match transform.

Option	Description
<i>Best record strategy</i>	<p>Choose the strategy to determine whether any action is taken on records in a match group. This is the criteria for further action. After you choose the strategy, priority, and field that you want to work with, the Match transform automatically generates the Python code for you (except for Custom).</p> <p><i>Custom</i>: Choose this strategy to base your strategy entirely on custom Python code. This allows you to open the Python Expression editor and create custom Python code.</p> <p><i>Date</i>: Choose Date to base your strategy on a date field.</p> <p><i>Length</i>: Choose Length to base your strategy on the length of data in a field.</p> <p><i>Non_Blank</i>: Choose Non_Blank to base your strategy on the completeness of data in a field.</p> <p><i>Priority_Number</i>: Choose Priority_Number to base your strategy on a number.</p> <p><i>Priority_String</i>: Choose Priority_String to base your strategy on string data in a field.</p>
<i>Strategy priority</i>	<p>These are the choices for priorities for each of the best record strategies, other than Non_Blank and Custom.</p> <p>Date</p> <ul style="list-style-type: none"> <i>Newest</i>: The newest date in the field will cause an action to take place. <i>Oldest</i>: The oldest date in a field will cause an action to take place. <p>Length</p> <ul style="list-style-type: none"> <i>Longest</i>: The longest string in a field will cause an action to take place. <i>Shortest</i>: The shortest string in a field will cause an action to take place. <p>Priority_Number</p> <ul style="list-style-type: none"> <i>Highest</i>: The highest number in a field will cause an action to take place. <i>Lowest</i>: The lowest number in a field will cause an action to take place. <p>Priority_String</p> <ul style="list-style-type: none"> <i>Ascending</i>: The string with the most ascending string order will cause an action to take place. <i>Descending</i>: The string with the most descending string order will cause an action to take place.
<i>Strategy field</i>	Choose a field that contains data that you need to execute your strategy.
<i>Posting destination</i>	<p>Specifies the destination record.</p> <p><i>Master</i>: Post only to a master record.</p> <p><i>Subs</i>: Post only to subordinate records.</p> <p><i>Master to Subs</i>: Push information from the master record and post it to each subordinate record.</p> <p><i>All</i>: Post to both the master and subordinate records.</p>

Option	Description
<i>Post only once per destination</i>	<p>Yes: Post only once per destination record. After data is posted to the destination record, the operation stops.</p> <p>No: Post more than once per destination. After data is posted to the destination record, the operation continues and the destination record is populated again with the next value. This option should be used when accumulating values such as total sales.</p> <p>Set this option to Yes when you are using the NON_BLANK strategy.</p> <p>Set this option to No when you are using the Longest, Shortest, Newest, Oldest, Ascending, or Descending priorities.</p> <div> <p>Note</p> <p>This option is ignored when using the <i>Master to Subs</i> posting destination. With this posting destination, information can be posted only once.</p> </div>
<i>View/Edit Python</i>	The View/Edit Python button opens the Python Expression editor. If you chose the Custom strategy, you can create your custom Python code. If you chose any other strategy, Python viewed in the editor is read-only.

Best record action fields table

Use the *Best record action fields* table to define the actions taken on the fields based on your strategy.

Option/Option group	Description
<i>Source field</i>	Specifies the name of the source field in the input record.
<i>Destination field</i>	<p>Specifies the name of the destination field in the output record, or the destination of this best record action.</p> <p>You can have the action post to the same input field, or you can post to a different field.</p>
<i>Custom</i>	<p>Yes: Specifies that you want to create custom Python code to perform an action on the destination field.</p> <p>No: Specifies that you want to use the same source and destination fields.</p> <p>When this option is set to No, the contents of the source field are copied to the destination field.</p>
<i>Editor</i>	If you chose Yes in the Custom column, a button appears here to allow you open the Python Expression editor and configure your Python code. You can open the Python Expression editor only if Custom is set to Yes.

8.5.11.4.4.2 Best record options: Destination Protection tab

Protect data from being changed by enabling and defining destination protection.

Option	Description
<i>Best record name</i>	Enter a name for your Best Record operation. Make sure that this name is unique within this Match transform.
<i>Enable destination protection</i>	Select to protect records from best record operations that may modify the contents.
<i>Default destination protection</i>	Select the default destination protection setting. This is useful because the default setting accounts for records that are protected (or not protected) through the use of sources or fields.
<i>Specify destination protection by field</i>	Select to enable destination protection through a value in a field.
<i>Destination protection field</i>	Choose the field that holds the destination protection value. The field must contain a value of Y or N. Any other value, including blank, causes the default destination protection specification to occur, if you specified a default destination protection.
<i>Specify destination protection by source</i>	Select this option to control destination protection through membership in a particular source. Fill in the table with source names and whether they are protected.
<i>Source Name</i>	Choose the name of the source from the dropdown list. The list here is populated with defined sources and source groups from the <i>Input Sources Editor</i> window of the Match Editor.
<i>Destination protected</i>	Select a value to assign to the source. Select Yes to enable destination for that source. Select No, if you do not wish to protect records from that source.

Parent topic: [Post-association processing \[page 457\]](#)

Related Information

[Best record options: Best Record tab \[page 459\]](#)

[Group Statistics Editor options \[page 463\]](#)

[Group prioritization options: Priority Order tab \[page 465\]](#)

[Group prioritization options: Record Completeness tab \[page 466\]](#)

[Unique ID options: Unique ID tab \[page 467\]](#)

[Unique ID options: Destination Protection tab \[page 470\]](#)

8.5.11.4.4.3 Group Statistics Editor options

Use the options in the *Group Statistics Editor* to set up an operation for post association processing.

Access the *Group Statistics Editor* by selecting the Associate transform in your data flow and selecting ► *Tools* ► *Associate Editor*. Then select Post Association Processing from the left pane and select *Add Operation* at right.

Option	Description
<i>Group statistics name</i>	Choose a name for this group statistics operation. If you are including more than one group statistics operation in this Match transform, make sure that the name is unique.
<i>Generate only basic statistics</i>	Select if you want to generate match group statistics. Basic statistics does not include any statistics about input sources.
<i>Generate source statistics from input sources</i>	Select to generate statistic counts about your input sources. You must have input sources defined in the Match editor for this option to be active. If you do not check this option, the Match transform still generates statistics about your match groups.
<i>Generate source statistics from source values</i>	Select to generate source statistics based on source values in a field. If you have a source value field, using this option, you can choose to count all sources or specific ones based on a particular value. When you select this option, the following options become editable: <ul style="list-style-type: none">• <i>Logical source field</i>• <i>Default logical source value</i>• <i>Count all sources</i>• <i>Choose sources to count</i>
<i>Logical source field</i>	Specifies the field that holds the value for your logical sources.
<i>Default logical source value</i>	Specifies a value to use if the field in the <i>Logical source</i> field option is blank. For example, if a record has a blank value in a field, the transform uses the value in the <i>Default logical source value</i> option.
<i>Count all sources</i>	Select to count all sources, no matter what the value in the <i>Logical source</i> field is.
<i>Choose sources to count</i>	Select to specify particular sources to count, based on values in the <i>Logical source</i> field. When you select this option, the following options become editable: <ul style="list-style-type: none">• <i>Default count flag</i>• <i>Auto generate sources</i>• <i>Predefined count flat field</i> (editable only when you select <i>Auto generate sources</i>)• <i>Manually define logical source count flags</i>

Option	Description
<i>Default count flag value</i>	<p>Specifies the value to use when the <i>Predefined count flag</i> field is invalid. For example, if the <i>Predefined count flag</i> field has data other than Y or N, or it is empty.</p> <p><i>Yes</i>: Counts all of the records in the source.</p> <p><i>No</i>: Does not count any of the records in the source.</p>
<i>Auto generate sources</i>	Select to generate sources based on the value in a field.
<i>Predefined count flag field</i>	Specifies the field name that contains the indicator value (Y or N) to determine whether a source is counted. This field is editable only when you also select <i>Auto generate sources</i> .
<i>Manually define logical source count flags</i>	
Be sure to fill in both columns for this option to work.	
<i>Source value</i>	Specifies the value in the field to find. This value is case sensitive.
<i>Count</i>	<p>Specifies whether you want to use the value you entered in the Logical source value option in the count.</p> <p><i>Yes</i>: Includes the logical source value in the count.</p> <p><i>No</i>: Does not include the value in the logical source value option in the count.</p>

Parent topic: [Post-association processing \[page 457\]](#)

Related Information

[Best record options: Best Record tab \[page 459\]](#)

[Best record options: Destination Protection tab \[page 462\]](#)

[Group prioritization options: Priority Order tab \[page 465\]](#)

[Group prioritization options: Record Completeness tab \[page 466\]](#)

[Unique ID options: Unique ID tab \[page 467\]](#)

[Unique ID options: Destination Protection tab \[page 470\]](#)

8.5.11.4.4.4 Group prioritization options: Priority Order tab

These options appear in both the *Match Editor* and the *Association Editor* based on which transform you are working with.

In the *Post Association Processing* or *Post Matching Processing* dialog of the *<transform_name> Editor*, select to add a Group Prioritization operation to control group order in post processing.

Group forming prioritization

Use the Group prioritization operation to order records within each break group. This order controls which records are used as the drivers during the comparison process.

Post-match prioritization

Add a Group prioritization operation before a Group Statistics operation to order records within a match group to control which record is flagged as the master record of each group of matching records. Add a Group prioritization operation before a Best Record operation to order records within a match group to control the destination of data that is being propagated from other records to form a best record.

Group Prioritization Editor option description

Option	Description
<i>Prioritization name</i>	Specifies the name for this Group prioritization operation. If you have multiple operations in this Match or Associate transform, be sure to make this name unique.

Priority fields

Use the Priority fields table to order your break groups based on the content of a field (for example, dollar amount or date). Use the buttons to add, remove, and order rows. Place the primary sort field at the top of the list. The rest of the fields, in the order that they are positioned, determine the sub-sort that occurs.

Option	Description
<i>Input field</i>	Choose a field to sort your records on.
<i>Field order</i>	Specifies in which order records should be sorted.

Parent topic: [Post-association processing \[page 457\]](#)

Related Information

Best record options: [Best Record tab \[page 459\]](#)

Best record options: [Destination Protection tab \[page 462\]](#)

Group Statistics Editor options [\[page 463\]](#)

Group prioritization options: [Record Completeness tab \[page 466\]](#)

Unique ID options: [Unique ID tab \[page 467\]](#)

Unique ID options: [Destination Protection tab \[page 470\]](#)

8.5.11.4.4.5 Group prioritization options: Record Completeness tab

Make settings to set the priority of output data based on record completeness.

Option	Description
<i>Prioritization name</i>	Specifies the name for this Group prioritization operation. If you have multiple operations in this Match transform, be sure to make this name unique.
<i>Order records based on completeness of data</i>	Select this option to apply priority and blank penalty points to records to help control the order of your records.
<i>Define only field penalties</i>	Select this option so the software assesses penalties based on blank fields.
<i>Define priority and penalty fields</i>	Select this option when you have specific fields that contain the actual integer values for priority and blank penalty.
<i>Record priority field</i>	Choose the field that contains priority values. This field must contain an integer.
<i>Apply blank penalty field</i>	Choose the field that contains the indicator (Y or N) for applying blank penalty points to a record.
<i>Define priority and penalty based on input source</i>	Select to have your record priority and blank penalty indicator (Y or N) determined by membership in a given source.
<i>Source Name</i>	Choose an input source from the dropdown list in the Source Name column. The sources listed here are defined in the Input Source operation.
<i>Priority</i>	Type a priority value (an integer) in the Priority column. Remember that the lower the priority score, the higher the priority.
<i>Apply Blank Penalty</i>	Choose Yes or No to determine whether a blank penalty is applied to a record based on membership to this source.
<i>Default record priority</i>	<p>Specifies the default value for the record priority if:</p> <ul style="list-style-type: none">• The record does not contain a field with this value.• The field is blank for a record.• A record does not belong to any of the sources specified. <p>Remember that the lower the priority score, the higher the priority.</p>
<i>Default apply blank penalty</i>	<p>Specifies the default indicator to add blank penalty points to records with blank fields. Use the indicator when a record does not have a field that carries this indicator, if that field is blank or has invalid data, or if a record does not belong to any of the sources specified.</p> <p>Yes: Each blank penalty for a record is added to the record priority to generate an adjusted record priority score. The lower the score, the higher the priority.</p> <p>No: The software does not apply a penalty when the fields are blank.</p>
<i>Input field</i>	Displays the input fields that are available to assign a blank penalty score to.
<i>Blank penalty</i>	Assign a penalty value (an integer) to apply when the specified field is blank in a record.

Parent topic: [Post-association processing \[page 457\]](#)

Related Information

Best record options: Best Record tab [page 459]

Best record options: Destination Protection tab [page 462]

Group Statistics Editor options [page 463]

Group prioritization options: Priority Order tab [page 465]

Unique ID options: Unique ID tab [page 467]

Unique ID options: Destination Protection tab [page 470]

8.5.11.4.4.6 Unique ID options: Unique ID tab

Use the Unique ID options to assign sequential identification numbers to each new record when adding records to a data warehouse.

For example, the largest number assigned in a particular project can be carried over as the beginning identification number plus 1. The software uses that beginning number in the assignment of new sequential ID numbers. The software assigns new sequential numbers when the software processes the next source against the data warehouse file.

i Note

Also see the Unique ID section for information about working with unique ID in a multiserver environment. Depending on the processing operation and starting value source you use, there could be limitations for using unique ID.

The Unique ID option group includes the following options:

Option	Description
<i>Unique ID name</i>	Enter a name for this Unique ID operation. If you are using other Unique ID operations in this Match transform, distinguish it from other operations. We suggest that you specify the name of the match transform and match level in the unique ID name.

Option	Description
<i>Processing operation</i>	<p>Specifies the type of processing operation you want the application to perform. Valid values include:</p> <p><i>Assign</i>: Assigns a new ID to unique records that need one, or assigns a new ID to all members of a group that don't have an ID. In addition, the assign operation copies an existing ID when a member of a match group already has an ID. For assign operations to work, all match group members must appear consecutively in one collection, and must be in priority order (high to low).</p> <p><i>AssignCombine</i>: Performs both an assign operation and a combine operation. All match group members must appear consecutively in one collection and must be in priority order (high to low).</p> <p><i>Combine</i>: Combines the IDs of a match group when more than one ID is represented. All match group members must appear consecutively in one collection and must be in priority order (high to low).</p> <p><i>Delete</i>: Removes unique IDs from records that have one, unless they are protected.</p> <p><i>Split</i>: Splits the IDs of an ID group when more than one match group is represented. All ID group members must appear consecutively in one collection and must be in priority order (high to low).</p>
<i>Recycle unique IDs</i>	<p>Specifies whether the software should reuse the unique IDs that were freed up during the delete operation in different records. If you have a limited amount of unique ID numbers available, you may want to recycle them. Valid values include:</p> <p><i>Yes</i>: Recycle freed-up unique IDs.</p> <p><i>No</i>: Do not recycle freed-up unique IDs.</p>
<i>ID field</i>	A field that holds a previously assigned unique ID. If this field is omitted, then it is assumed that no records have a unique ID.
<i>Field</i>	<p>The software obtains the starting unique ID from an input field.</p> <p>Be sure to map in a field from an upstream transform before you add this option.</p>
<i>Starting unique ID field</i>	Choose the field that passes in the starting unique ID. If no Unique ID is received, the starting number defaults to 1.
<i>Constant value</i>	The starting ID is specified as a positive whole number in the Starting value option.
<i>Starting value</i>	Indicates the starting unique ID value. Valid values range from 1 to UINT_MAX (unsigned integer max). The default value is 1.
<i>Value from file</i>	The starting Unique ID is read from the file specified in the File option.
<i>File</i>	Specifies the path and name of the file that manages unique IDs. A value is required here only when the Starting unique ID source option is set to File.
<i>GUID</i>	<p>Uses the Globally Unique Identifier (GUID) as the unique ID. GUID is also known as the Universal Unique Identifier (UUID). The UUID variation used for unique ID is a time-based 36-character string with the format: TimeLow-TimeMid-TimeHighAndVersion-ClockSeqAndReservedClockSeqLow-Node.</p> <p>For more information about UUID, see the RFC document in the Related Topics section.</p>

Option	Description
<i>Save ending ID to file and reclaim recycled IDs</i>	Specifies whether to save the last ID that was assigned to a file. Additionally, specifies whether to reclaim recycled IDs.
<i>File</i>	Specifies the file to write the last assigned ID to.
<i>Allow multiple Match transforms to access unique ID file</i>	Allows multiple Match transforms to access a shared unique ID file. When enabled, multiple data flows can access the same unique ID file, and single Match transforms can run in more than one process when the DOP setting is greater than 1. In addition, this option allows multiple Match transforms within a single data flow to share a single unique ID file.
<i>Number of IDs to get when accessing file</i>	Specifies the number of IDs to retrieve from the unique ID file during each access. For example, with a setting of 100, the first process accesses the file and retrieve IDs numbered 1-100. The next process with retrieve IDs numbered 101-200. If a process uses less than the number of retrieved IDs, the remaining IDs are written back to the unique ID file as recycled IDs.
<div>i Note</div> <p>A setting greater than 1 improves performance when sharing a unique ID file between multiple processes by reducing the number of times the file must be accessed. However, integer numbers may not be assigned in sequential order.</p>	
<i>Group number field</i>	Specifies the field that holds a group number. The group number is used to assign the same unique ID to more than one record. If this field is omitted, then it is assumed that each record is unique and should have its own number.

Parent topic: [Post-association processing \[page 457\]](#)

Related Information

Best record options: [Best Record tab \[page 459\]](#)

Best record options: [Destination Protection tab \[page 462\]](#)

Group Statistics Editor options [\[page 463\]](#)

Group prioritization options: [Priority Order tab \[page 465\]](#)

Group prioritization options: [Record Completeness tab \[page 466\]](#)

Unique ID options: [Destination Protection tab \[page 470\]](#)

8.5.11.4.4.7 Unique ID options: Destination Protection tab

Use the Destination Protection tab to control whether a record unique ID is protected based on the source that the record belongs to.

This protection can help prevent IDs from being assigned to a suppression or rented source.

Option	Description
<i>Unique ID name</i>	Enter a name for this Unique ID operation. If you are using other Unique ID operations in this Match transform, you may want to specify the name of the match transform and match level in this name to distinguish it from others.
<i>Enable destination protection</i>	Select if you want to protect a destination source from having its unique IDs overwritten with the IDs from matching records.
<i>Default destination protection</i>	Select the default destination protection setting. The protection setting is useful because the default setting accounts for records that are protected (or not protected) through the use of sources or fields.
<i>Specify destination protection by field</i>	Select to enable destination protection through a value in a field.
<i>Unique ID protected field</i>	Choose an input field from the dropdown list that holds the value for specifying whether this ID is protected. The field must contain a value of Y or N. Any other value (including blank) causes the default destination protection specification to occur, if you specified one.
<i>Specify destination protection by source</i>	Select this option to control destination protection through membership in a particular source. Fill in the table with source names and whether they are protected.
<i>Source name</i>	Choose the name of the source from the dropdown list. The list here is populated with defined sources and source groups from the Input Sources Editor window of the Match Editor.
<i>Unique ID protected</i>	<p>Yes: The source is protected.</p> <p>No: The source is not protected.</p>

Parent topic: [Post-association processing \[page 457\]](#)

Related Information

Best record options: [Best Record tab \[page 459\]](#)

Best record options: [Destination Protection tab \[page 462\]](#)

Group Statistics Editor options [\[page 463\]](#)

Group prioritization options: [Priority Order tab \[page 465\]](#)

Group prioritization options: [Record Completeness tab \[page 466\]](#)

Unique ID options: [Unique ID tab \[page 467\]](#)

8.5.11.4.4.8 Output flag selection options

Select types of records you want to flag on output based on each of the input sources. You may want to flag these records so that they will be available for writing to output. Downstream in a data flow, you can check the value of the Select_Record (Y/N) and decide whether you want to write it to output by using a Case transform, for example.

This is a repeatable operation.

Record type	Description
<i>Output flag selection name</i>	<p>Enter a unique name for this operation that will allow you to identify it in a report and in your Select_Record output field.</p> <p>For example, suppose you have two two Output Flag Selection operations in this match level: DMA_Matches and Mail_List. Your output fields are then called:</p> <ul style="list-style-type: none">• <match level name>_DMA_Matches_Select_Record• <match level name>_Mail_List_Select_Record
<i>Unique</i>	Records that are not members of any match group. No matching records were found. These can be from sources with a normal- or special-type source.
<i>Single source masters</i>	Highest ranking member of a match group whose members all came from the same source. Can be from normal- or special-type sources.
<i>Single source subordinates</i>	A record that came from a normal- or suppress-type source and is a subordinate member of a match group.
<i>Multiple source masters</i>	Highest ranking member of a match group whose members came from two or more sources. Can be from normal- or special-type sources.
<i>Multiple source subordinates</i>	A subordinate record of a match group that came from a normal- or suppress-type source whose members came from two or more sources.
<i>Suppression matches</i>	Subordinate member of a match group that includes a higher-priority record that came from a suppress -type source. Can be from normal- or special-type source.
<i>Suppression uniques</i>	Records that came from a suppress-type source, and for which no matching records were found.
<i>Suppression masters</i>	A record that came from a suppress-type source and is the highest ranking member of a match group.
<i>Suppression subordinates</i>	A record that came from a suppress-type source and is a subordinate member of a match group.

8.5.11.5 Input fields for the Match transform

The following are recognized input fields that you can use in the input mapping for the Match transform.

Input field name (Match)	Description
Option_Field_Algorithm_Geo_Proximity_<logical_name>_Max_Distance	<p>The distance used in proximity matching.</p> <p>This setting is dynamic. If you change this setting, you do not have to terminate and reinitialize the transform in order for the new configuration to be recognized.</p> <p><logical-name> is a name used to reference option groups. Some option groups can be repeated and the transform needs a way to uniquely identify each option group.</p> <p>In the Designer, this name is generated automatically by the Match editor. In order to understand which option is affected by this input field, look in the option tab of the Match transform and find the Field Algorithm Geo Proximity option group whose name is the same as appears in the input field.</p>

8.5.11.6 Output fields for the Match transform

The following Match fields are generated by the Match transform per match level. Use these fields when you map your output schema.

Output field name (Match)	Default content type	Description
All_Match_Criteria	None	For internal use only
All_Match_Records	None	For internal use only
Break_Key	None	Specifies the break key generated for this record.

Output field name (Match)	Default content type	Description
Conflict_Record	None	<p>The sequence number of the record that has a conflict with this record. Conflict means that records that do not match directly are in the same match group.</p> <p>For example, records A and B match record C, but records A and B do not match directly, therefore the pair of records A and B are conflict records.</p> <p>For records that have no conflict, the value is blank.</p> <div> <p>i Note</p> <p>Select this field for output only if your job was created by Information Steward Data Cleansing Advisor and you have the Match transform configured for pattern matching (which is not currently available). Using this field under other conditions will result in output that is either blank or provides no value.</p> </div>
First_Match_Record	None	<p>Specifies the sequence number of the record that this record first matched. Unique records will have a value of Null. This output field is useful when <i>Match Mode</i> is set to "Multi-Driver" to determine to which record a record first matched.</p> <div> <p>i Note</p> <p>Select this field for output only if your job was created by Information Steward Data Cleansing Advisor and you have the Match transform configured for pattern matching (which is not currently available). Using these fields under other conditions will result in blank output.</p> </div> <div> <p>i Note</p> <p>Currently, the Match mode option is not available for edit in Data Services.</p> </div>

Output field name (Match)	Default content type	Description
Group_Number	Group_Number	Specifies the records that belong to the same match group, which share the same group number. The group numbers start with the number 1. Unique records have a blank group number. If you are using association matching in your job, you need to map this on output, because the Associate transform uses it.
Match_Criterion	None	Specifies the name of the criteria or pattern that made the decision (if the Match_Type is <i>R</i>). Otherwise, the field is blank. <div> i Note Pattern matching is not currently available in Data Services. </div>
Match_Level	None	Specifies the name of the match level used.
Match_Pair_Count	None	Specifies how many records this record matched. A unique record will have a value of 0. A record that matched all the other records in a match group will have a value that is one less than the match group count. If Match Mode is set to "Single Driver", then the driver record will have a match pair count equaling one less than the match group count, and the passengers will have a match pair count of 1. <div> i Note Select this field for output only if your job was created by Information Steward Data Cleansing Advisor and you have the Match transform configured for pattern matching (which is not currently available). Using this field under other conditions will result in output that is either blank or provides no value. </div> <div> i Note Currently, the Match mode option is not available for edit in Data Services. </div>

Output field name (Match)	Default content type	Description
Match_Score	None	<p>The Match_Score field outputs the following values:</p> <ul style="list-style-type: none"> • The criteria or pattern similarity score when the Match_Type is <i>R</i>. • The total weighted score when the Match_Type is <i>W</i>. • Blank if the record is a driver record (Match_Type of <i>D</i>) or if the records are unique. <div> i Note Pattern matching is not currently available in Data Services. </div>
Match_Status	None	<p>The values for the Match_Status field that appear in your output are:</p> <p><i>D</i>: This record is a driver in a match group.</p> <p><i>P</i>: This record is a passenger in a match group.</p> <p><i>U</i>: This is a unique record.</p> <p><i>B</i>: Signifies that the <i>Match mode</i> option is set to “Multi-Driver” and the record is both a driver and passenger.</p> <div> i Note Currently, the Match mode option is not available for edit in Data Services. </div>

Output field name (Match)	Default content type	Description
Match_Type	None	<p>Indicates what type of match was made to bring the record into a match group. Possible values are:</p> <p><i>blank</i>: The record did not match any other record. It is a unique record.</p> <p><i>D</i>: The record was the driver record in the comparison process.</p> <p><i>R</i>: The decision was made by a rule (criteria) or pattern.</p> <p><i>W</i>: The record was identified as matching the driver because the total weighted score met the Weighted match score. See "Matching methods" in the <i>Data Services Designer Guide</i> for more information on weighted scores.</p> <div> <p>i Note</p> <p>Pattern matching is not currently available in Data Services.</p> </div>
Review_Record	None	<p>The sequence number of the record that matches with this record but something about the pair of matching records is identified as questionable.</p> <p>For records that do not need review, the value is blank.</p> <div> <p>i Note</p> <p>Select this field for output only if your job was created by Information Steward Data Cleansing Advisor and you have the Match transform configured for pattern matching (which is not currently available). Using this field under other conditions will result in output that is either blank or provides no value.</p> </div>

Output field name (Match)	Default content type	Description
Sequence_Number	None	<p>Each record input to the Match transform is assigned a sequence number. This number will be stored internally by a 64-bit integer. The first value assigned after the Match transform is initialized is 1, the second value is 2, and so on. The sequence number will allow Match to connect two records by having them reference each other's sequence number. The field data type is Decimal.</p> <div> <p>Note</p> <p>Select this field for output only if your job was created by Information Steward Data Cleansing Advisor and you have the Match transform configured for pattern matching (which is not currently available). Using this field under other conditions will result in output that is either blank or provides no value.</p> </div>

Input Source operation output fields

These fields are available only when you use an Input Source operation in your Match transform.

Output field name (Match-Input Source)	Description
Source_Group_Name	Specifies the name of the Source Group that the current record belongs to. If a record does not belong to any Source Group, then an empty string is output.
Source_Name	Specifies the name of the input source that the current record belongs to.
Source_Type	<p>Specifies the type of source that the current record belongs to.</p> <p><i>N</i>: The record comes from a normal source.</p> <p><i>P</i>: The record comes from a suppress source.</p> <p><i>S</i>: The record comes from a special source.</p>

If you also add a Group Statistics post-match operation, and select the [Generate source statistics from input sources](#) option, the following output fields are available (these are in addition to the fields generated by the Group statistics operation).

Output field name (Match–Input Source)	Description
Group_Count	Provides the total number of records in the match group. Unique records have a value of 1.
Group_Rank	Specifies whether the record is a master (M) or a subordinate (S). Unique records have an empty value.
Group_Source_Appearance	Specifies the order the input source appears in this match group. The first input source appearing in the match group receives a value of 1, the second Input Source appearing will get 2, and so forth. Records that come from the same input source will receive the same Group_Source_Appearance value. Unique records have a value of 0.
Group_Source_Group_Order	Specifies the order of the records within the match group that have the same Group_Source_Group_Appearance value. The first occurrence receives a value of 1, the second occurrence receives a value of 2, and so on. Unique records receive a value of 0. Records in a match group not assigned to a source group will also get a value of 0.
Group_Source_Order	Specifies the order of the records within the match group that have the same Group_Source_Appearance value. The first occurrence receives a value of 1, the second occurrence receives a value of 2, and so on. Unique records have a value of 0.
Group_Source_Type	Specifies the type of source in the match group. This field will contain one of the following values: <i>M</i> : The records come from more than one input source (excluding records from Special sources). <i>P</i> : The records come from a Suppress source. (If the master record comes from a suppression source, then all records in the match group have a P. If the master record comes from a normal or special source, then the suppression record and all records after it have a P, but the records before the suppression record have an M or S.) <i>S</i> : The records come from a single input source . <i>empty</i> : The record is unique.
Multi_Source_Count	Specifies the number of sources represented in the match group (excluding the Special sources and Suppress sources and Normal sources that follow a Suppress source in the match group). The values of this field could range from 0 to the number of records in the match group. Unique records receive a value of 1, if from a Normal list, and 0, if from a Special source or a Suppress source.
Source_Count	Specifies the number of sources represented in the match group (regardless of the source types). The values of this output field could range from 1 to the number of records in the match group. Unique records will have a value of 1.

Output field name (Match–Input Source)	Description
Source_Group_Count	Specifies the number of Source Groups represented in the match group. Records in the match group that do not belong to a Source Group are not counted. The value of this output field range from 0 to the maximum number of input sources (10,000). Unique records will have a value of 0 or 1.

If you also add a Group Statistics post-match operation, and select the [Generate source statistics from source values](#) option, the following output fields are available (these are in addition to the fields generated by the Group statistics operation).

Output field name (Match–Input Source)	Description
Group_Count	Provides the total number of records in the match group. Unique records have a value of 1.
Group_Order	The master record receives a value of 1. Subordinate records receive a value of 2 through the number of records in the match group. You may control the order by including a Group Prioritization in the Post Match Operations. Unique records have a value of 0.
Group_Rank	Specifies whether the record is a master (M) or a subordinate (S). Unique records have an empty value.
Group_Type	Specifies whether a record contributed to the source count, and if so, whether there were other sources represented in the match group. M: Multiple sources. Records from multiple sources are represented in the match group (records from Special sources are not counted toward a multiple-source match group). S: Single source. All records in the match group come from a single source (records from Special sources are not counted toward a multiple-source match group). P: At least one record from a Suppression source is included in the match group. (If the master record comes from a suppression source, then all records in the match group have a P. If the master record comes from a normal or special source, then the suppression record and all records after it have a P, but the records before the suppression record have an M or S.)
Source_Count	Specifies the number of sources represented in the match group (regardless of the source types). The values of this output field could range from 1 to the number of records in the match group. Unique records will have a value of 1.
Source_ID	Specifies the logical source value. In most cases, this is the input source value. In other cases it is the default logical source value.

Output field name (Match–Input Source)	Description
Source_ID_Count	Specifies the number of source IDs represented in the match group.
Source_Type_ID	Specifies the type of logical source. <i>N</i> : Normal source <i>P</i> : Suppress source <i>S</i> : Special source

Source Group operation output fields

These fields are available only if you use a Source Group operation in your Match transform.

Output field name (Match–Source Group)	Description
Group_Source_Group_Appearance	Specifies the order the source group appears in this match group. The first source group appearing in the match group receives a value of 1, the second source group appearing receives a value of 2, and so on. Records that come from the same source group will receive the same Group_Source_Group_Appearance value. Unique records receive a value of 0. Records in a match group not assigned to a source group will also get a value of 0.
Group_Source_Group_Order	Specifies the order of the records within the match group that have the same Group_Source_Group_Appearance value. The first occurrence receives a value of 1, the second occurrence receives a value of 2, and so on. Unique records receive a value of 0. Records in a match group not assigned to a source group will also get a value of 0.
Source_Group_Count	Specifies the number of source groups represented in the match group. Records in the match group that do not belong to a source group are not counted. The values of this output field could range from 0 to the number of records in the match group. Unique records receive a value of 0 or 1.
Source_Group_Name	Specifies the name of the source group that the current record belongs to. If a record does not belong to any source group, then an empty string is output.

Group Statistics operation output fields

These fields are available only if you use a Group Statistics operation in your Match transform.

Output field name (Match–Group Statistics)	Description
Conflict_Group	<p>Contains C if the record is part of a match group that includes a conflict. Conflict means that records that do not match directly are in the same match group.</p> <p>For example, records A and B match record C, but records A and B do not match directly, therefore the pair of records A and B are conflict records.</p> <p>For records that do not belong to a match group with any conflicts, the value is blank.</p> <div> <p>i Note</p> <p>Select this field for output only if your job was created by Information Steward Data Cleansing Advisor and you have the Match transform configured for pattern matching (which is not currently available). Using this field under other conditions will result in output that is either blank or provides no value.</p> </div>
Group_Count	<p>Provides the total number of records in the match group.</p> <p>Unique records have a value of 1.</p>
Group_Order	<p>The master record receives a value of 1. Subordinate records receive a value of 2 through the number of records in the match group.</p> <p>You may control the order by including a Group Prioritization in the Post Match Operations. Unique records have a value of 0.</p>
Group_Rank	<p>Specifies whether the record is a master (M) or a subordinate (S). Unique records have an empty value.</p>
Group_Type	<p>Specifies whether a record contributed to the source count, and if so, whether there were other sources represented in the match group.</p> <p>M: Multiple sources. Records from multiple sources are represented in the match group (records from Special sources are not counted toward a multiple-source match group).</p> <p>S: Single source. All records in the match group come from a single source (records from Special sources are not counted toward a multiple-source match group).</p> <p>P: At least one record from a Suppression source is included in the match group. (If the master record comes from a suppression source, then all records in the match group have a P. If the master record comes from a normal or special source, then the suppression record and all records after it have a P, but the records before the suppression record have an M or S.)</p>

Output field name (Match–Group Statistics)	Description
Review_Group	<p>Contains R if the record is part of a match group for which review is recommended. Review is recommended because the pair of matching records is identified as questionable. For records that do not belong to a match group with any review records, the value is blank.</p> <div> i Note <p>Select this field for output only if your job was created by Information Steward Data Cleansing Advisor and you have the Match transform configured for pattern matching (which is not currently available). Using this field under other conditions will result in output that is either blank or provides no value.</p> </div>
Source_Count	<p>Shows the number of logical sources in this match group.</p> <p>Unique records have a blank value</p>
Source_ID	<p>Specifies the logical source value. In most cases, this is the input source value. In other cases it is the default logical source value.</p>
Source_ID_Count	<p>Specifies the number of source IDs represented in the match group.</p>
Source_Type_ID	<p>Specifies the type of logical source.</p> <p><i>N</i>: Normal source</p> <p><i>P</i>: Suppress source</p> <p><i>S</i>: Special source</p>

Unique ID operation output fields

These fields are available only if you use a Unique ID operation in your Match transform.

Output field name (Match–Unique ID)	Description
ID_Status	<p>Specifies the status of the Unique_ID output field. This field generates the following values:</p> <p><i>blank</i>: No change. The Unique_ID output field is the same as the Unique_ID input field.</p> <p><i>D</i>: Indicates that the Unique_ID output field has a blank unique ID and that the old unique ID was deleted.</p> <p><i>N</i>: Indicates that the Unique_ID output field has a new unique ID.</p> <p><i>O</i>: Indicates that the Unique_ID output field is assigned an old (existing) ID. This happens when a record is combined.</p>
Record_Operation	<p>Specifies the operation that should be performed on the record, based on the input fields (except protected fields). This field generates the following values:</p> <p><i>blank</i>: Does not call an operation.</p> <p><i>A</i>: Assigns a unique ID to the record.</p> <p><i>C</i>: Combines the record's unique ID.</p> <p><i>D</i>: Deletes the record's unique ID.</p> <p><i>S</i>: Splits the record's unique ID.</p>
Unique_ID	<p>Specifies the unique ID the Match transform generated for this record. If the record already has a valid unique ID, then the output field will output the same unique ID. If the Match transform does not assign a unique ID, the output field is blank.</p>

Group Prioritization output fields

The following output fields are available when you add a Group Prioritization operation to a Match transform

Output field name (Match–Group Prioritization)	Description
Priority_Value	<p>The sum of all priority and blank penalty values defined in the Record Completeness tab of the Group Prioritization. If you do not order records using the Record Completeness tab, this field contains 0.</p>

Output flag selection output fields

The following output fields are available when you add an Output flag election operation to a Match transform.

Output field name (Match–Output flag)	Description
Select_Record	Specifies whether the current record should be selected or not, based on your selections in the Output Flag Selection Editor. Valid values of this output field are Y if the record should be selected and N if the record should not be selected.

8.5.12 USA Regulatory Address Cleanse



The USA Regulatory Address Cleanse transform identifies, parses, validates, and corrects U. S. address data according to the U.S. Coding Accuracy Support System (CASS). This transform can create the USPS Form 3553 and output many useful codes to your records. You can also run in a non-certification mode as well as produce suggestion lists.

i Note

If an input record has characters not included in the Latin1 code page, the USA Regulatory Address Cleanse transform will not process that data. Instead, the software sends the mapped input record to the corresponding standardized output field (if applicable). No other output fields will be populated for that record. If your Unicode database has valid U.S. addresses from the Latin1 character set, the transform processes as normal.

If you perform both data cleansing and matching, the USA Regulatory Address Cleanse transform typically comes before the Data Cleanse transform and any of the Match transforms in the data flow. SAP recommends using a sample job or data flow that is set up according to best practices for a specific use case.

The USA Regulatory Address Cleanse transform has several sample transform configurations that can help you set up your data flow. The transforms include all of the required options except input fields.

Parent topic: [Data Quality transforms \[page 437\]](#)

Related Information

[Blueprints and other content objects for download \[page 438\]](#)

[About Data Quality fields \[page 445\]](#)

[About data quality statistics \[page 448\]](#)

[Associate \[page 454\]](#)

[Country ID \[page 472\]](#)

[Data Cleanse \[page 474\]](#)

[DSF2® Walk Sequencer \[page 514\]](#)
[Geocoder \[page 524\]](#)
[Global Address Cleanse \[page 552\]](#)
[Global Suggestion List \[page 640\]](#)
[Match \[page 651\]](#)
[Address Cleanse reference \[page 765\]](#)
[Data Cleanse reference \[page 823\]](#)
[Address Cleanse reference \[page 765\]](#)
[Transform configurations \[page 440\]](#)

8.5.12.1 Content objects

We provide content objects to help you create and complete transforms.

Transform configurations

A transform configuration is a transform with preconfigured input fields, output fields, and options that can be used in multiple data flows. These are useful if you repeatedly use a transform with specific options and input and output fields.

When Data Services is installed, read-only transform configurations are provided for the Data Quality transforms.

You can use transform configurations in your data flows or as an example of a typical transform. After you place an instance of the transform configuration in a data flow, you can override these preset defaults. You can also create your own transform configuration, either by replicating an existing transform configuration or creating a new one.

Sample blueprints and other objects

We have created Data Quality blueprints and other content objects to help you set up Data Services jobs. We've identified a number of common scenarios that you are likely to perform with Data Services. For each scenario, we've included a blueprint that is already set up to solve the business problem in that scenario.

Related Information

[Downloading blueprints and other content objects \[page 444\]](#)
[Transform configurations \[page 440\]](#)

8.5.12.2 Option groups

The USA Regulatory Address Cleanse transform includes options that you complete to process address data from the United States. The option groups are listed in the Related Topics list below:

Related Information

[Report And Analysis \[page 712\]](#)
[Reference Files \[page 713\]](#)
[Transform Performance \[page 715\]](#)
[USPS license information options \[page 716\]](#)
[NCOALink options \[page 719\]](#)
[Assignment Options \[page 726\]](#)
[Standardization options \[page 728\]](#)
[Non Certified Options \[page 736\]](#)
[CASS Report Options \[page 738\]](#)
[Suggestion List group \[page 739\]](#)
[Z4_Change_Options \[page 742\]](#)

8.5.12.3 Report and analysis

Use these options to generate USA Regulatory Address Cleanse report data.

Option	Description
Gather Statistics Per Data Source	<p>Specifies whether to generate report data per Data_Source_ID field value.</p> <p>Yes: Generates report statistics (if the Generate Report Data option is set to Yes) per Data_Source_ID field value.</p> <p>No: Generates reports (if the Generate Report Data option is set to Yes) for the input database without generating statistics based on the Data_Source_ID field value.</p> <div><p>i Note</p><p>If you select Yes, other setup requirements apply. Read about statistics based on logical groups in the <i>Designer Guide</i>.</p></div>
Generate Report Data	<p>Specifies whether to generate report data for this transform.</p> <p>Yes: Generates report data for this transform.</p> <p>No: Turns off report data generation. If you do not need to generate reports (during testing, for example), you should set this option to No to improve performance.</p>

8.5.12.4 Reference files

Reference files are directories used by the USA Regulatory Address Cleanse transform to correct and standardize your data. It is best to use the applicable substitution variable for the Option Value. This substitution variable represents the path to the reference files, and if you change that location, you can change the substitution variable so that all of your jobs reflect the current location.

Option	Description
Address Directory 1	<code>zip4us.dir</code> This directory, also called the National Directory, is organized by ZIP Code. It lists street names, ranges of house numbers, and postal and other codes. Use the substitution variable <code>\$\$RefFilesAddressCleanse</code> .
Address Directory 2	<code>*.dir</code> This second address directory is optional, and can be used for a customized address directory. No directory is automatically provided for this option. Most users should leave the Address Directory 2 option blank.
Address Geo 1 Directory	<code>ageo1.dir</code>
Address Geo 2 Directory	The Address-level GeoCensus files, <code>ageo1.dir</code> through <code>ageo10.dir</code> , are required only if you use the Address-level GeoCensus option or if you use the Geocoder transform. Use the substitution variable <code>\$\$RefFilesAddressCleanse</code> .
Address Geo 3 Directory	
Address Geo 4 Directory	
Address Geo 5 Directory	
Address Geo 6 Directory	
Address Geo 7 Directory	
Address Geo 8 Directory	
Address Geo 9 Directory	
Address Geo 10 Directory	
Address SHS Directory	<code>zip4us.shs</code> This directory enhances normal primary name lookups and is required for processing. Use the substitution variable <code>\$RefFilesAddressCleanse</code> .
Centroid Geo Directory	<code>cgeo2.dir</code> This directory is required only if you use the centroid-level GeoCensus option or if you use the Geocoder transform. Use the substitution variable <code>\$\$RefFilesAddressCleanse</code> .

Option	Description
<i>City Directory</i>	<p>city##.dir</p> <p>The City directory is a table of city names, states, and ZIP Codes. It is organized by state and city. Use the substitution variable <code>\$\$RefFilesAddressCleanse</code>.</p>
<i>DPV Path</i>	<p>Specify the path to the DPV (Delivery Point Validation) directory files. These directory files are required for CASS certification. Use the substitution variable <code>\$\$RefFilesAddressCleanse</code>.</p>
<i>DSF2 Augment Path</i>	<p>Specify the path to the directory that contains the DSF2 (Second Generation Delivery Sequence) files you received from the USPS.</p> <p>Use the substitution variable <code>\$\$RefFilesAddressCleanse</code>.</p>
<i>eLOT Directory</i>	<p>elot.dir</p> <p>The eLOT directory contains eLOT codes for the delivery point that represents the mail carrier's delivery route walk sequence.</p> <p>Include this directory only if the <i>Enable eLot</i> option in the Assignment Options group is set to <i>Yes</i>.</p> <p>Use the substitution variable <code>\$\$RefFilesAddressCleanse</code>.</p>
<i>EWS Directory</i>	<p>ewsyymmdd.dir</p> <p>The software lists <i>ew*.dir</i> in the Option Value column, so that the transform finds the most current directory.</p> <p>Use the substitution variable <code>\$\$RefFilesAddressCleanse</code>.</p>
<i>LACSLink Path</i>	<p>Specify the path to the LACSLink directory files. These directory files are required for CASS certification.</p> <p>Use the substitution variable <code>\$\$RefFilesAddressCleanse</code>.</p>
<i>NCOALink Path</i>	<p>The location for the NCOALink directory files. These directory files are required for NCOALink processing and certification.</p> <p>Use the substitution variable <code>\$\$RefFilesAddressCleanse</code>.</p>
<i>Postcode Directory</i>	<p>zcf10.dir</p> <p>This directory contains the same data as the City directory, but is organized by ZIP Code.</p> <p>Use the substitution variable <code>\$\$RefFilesAddressCleanse</code>.</p>
<i>Postcode Reverse Directory</i>	<p>revzip4.dir</p> <p>The Reverse ZIP+4 directory enables the software to assign more postal codes when the input data includes a unique ZIP Code and valid ZIP+4.</p> <p>Use the substitution variable <code>\$\$RefFilesAddressCleanse</code>.</p>
<i>RDI Path</i>	<p>The RDI directory indicates whether an address is residential.</p> <p>Use the substitution variable <code>\$\$RefFilesAddressCleanse</code>.</p>

Option	Description
<i>Reverse Soundex Address Directory</i>	<p>zip4us.rev</p> <p>This directory enhances primary name lookups.</p> <p>Use the substitution variable \$\$RefFilesAddressCleanse.</p>
<i>SuiteLink Path</i>	<p>The SuiteLink directories contain specially indexed address information such as secondary numbers and unit designators for locations identified as high-rise business default buildings. These directory files are required for CASS certification.</p> <p>Specify the path to the SuiteLink directory files.</p> <p>Use the substitution variable \$\$RefFilesAddressCleanse.</p>
<i>USPS Log Path</i>	<p>Specify the path to the directory for NCOALink, DPV, and LACSLink log files. The provided substitution parameter is \$\$CertificationLogPath. The software determines the file names during processing as the USPS requires. This directory must already exist and be writable.</p> <p>It is important to use the same path for all jobs. If you have multiple clients, use the same log file directory for all clients so that the log files are combined.</p>
<i>Z4 Change Directory</i>	<p>z4change.dir</p> <p>The Z4Change directory lists all the ZIP Codes and ZIP+4 Codes in the country.</p> <p>Use the substitution variable \$\$RefFilesAddressCleanse.</p>

For information about downloading directories, see the latest directories update.

8.5.12.5 Transform performance

The Transform Performance option group for the USA Regulatory Address Cleanse transform contains options that could improve the performance of DPV, DSF2, LACSLink, NCOALink, RDI, and SuiteLink processing.

Option	Description
<i>Cache DPV Directories</i>	<p>Specifies whether the DPV directories are cached into memory. If the directories are cached, the caching takes place only once and is shared among all DPV threads running in the data flow.</p> <p><i>Yes:</i> Enables caching.</p> <p><i>No:</i> Disables caching.</p>
<i>Cache DSF2 Augment Directories</i>	<p>Specifies whether the DSF2 Augment directories are cached into memory.</p> <p><i>Yes:</i> Enables caching.</p> <p><i>No:</i> Disables caching.</p>

Option	Description
<i>Cache LACSLink Directories</i>	<p>Specifies whether the LACSLink directories are cached into memory.</p> <p><i>Yes</i>: Enables caching.</p> <p><i>No</i>: Disables caching.</p>
<i>Cache RDI Directories</i>	<p>Specifies whether the RDI directories are cached into memory.</p> <p><i>Yes</i>: Enables caching.</p> <p><i>No</i>: Disables caching.</p>
<i>Cache SuiteLink Directories</i>	<p>Specifies whether the SuiteLink directories are cached into memory.</p> <p><i>Yes</i>: Enables caching.</p> <p><i>No</i>: Disables caching.</p>
<i>Insufficient Cache Memory Action</i>	<p>Specifies the action to take if there is insufficient memory to cache the directories that you have set up for caching.</p> <p><i>Continue</i>: Attempts to continue initialization without caching.</p> <p><i>Error</i>: Issues an error and terminates the transform.</p>
<i>NCOALink Caching Mode</i>	<p>Specifies the method for caching NCOALink directories.</p> <p><i>Auto</i>: Select to have Data Services use available memory for caching.</p> <p><i>Manual</i>: Select if you have a limited amount of memory available and you want to allocate a set amount of memory for caching. Enter a value in the <i>NCOALink Memory Usage</i> option.</p> <p><i>None</i>: Disables caching. This is the default setting. Consider this option for smaller jobs because the overhead of caching directories could take longer than the actual job execution duration.</p>
<i>NCOALink Memory Usage</i>	<p>If the <i>NCOALink Caching Mode</i> is set to Manual, enter a value here to allocate a set amount of memory for NCOALink directory caching. The transform uses this value as the maximum amount of memory that can be used for the caching of NCOALink directories.</p>


i Note

If the *Degree of Parallelism* value is greater than one, the *NCOALink Memory Usage* value is the total to be allocated for all threads. The value is not per thread.

8.5.12.6 USPS license information options

This group of options is required for all users performing NCOALink, SuiteLink, LACSLink, DPV, and DSF2 processing. You must provide information about the company performing the processing (the licensee) and the company for whom they are performing the processing (the customer). If you are performing the processing for yourself, you are the licensee and the customer.

The following table describes the *USPS Licensee and Customer Information Options* for the USA Regulatory Address Cleanse transform.

Option	Description
<i>Customer Company Name</i>	The customer is the person or company for whom you are performing NCOALink processing. End users may leave these fields blank unless you have an alternate stop processing agreement and have entered the special keycode into License Manager.
<i>Customer Company Address</i>	The customer information appears in the NCOALink Processing Summary report and log files.
<i>Customer Company Locality</i>	The provided substitution parameters for these fields are: \$\$CompanyName
<i>Customer Company Region</i>	\$\$CompanyAddress
<i>Customer Company Postcode1</i>	\$\$CompanyLocality
<i>Customer Company Postcode2</i>	\$\$CompanyRegion
	\$\$CompanyPostcode1
	\$\$CompanyPostcode2
<i>Customer Company Phone</i>	This is an optional field. The provided substitution parameter for this field is \$ \$CompanyPhone.
<i>DSF2 Licensee ID</i>	Enter your DSF2 identification number as the USPS assigned it to you. You can use the substitution variable \$\$DSF2LicenseeID in this option.
<i>IMB Mailer ID</i>	This is an optional field. Enter your unique Intelligent Mail barcode (IMB) mailer ID that you received from the USPS, if applicable. The provided substitution parameter for this field is \$\$IMBMailerID. The IMB Mailer ID is a unique 6-digit or 9-digit numeric code assigned to mailers by the USPS based on their annual mail volumes. This information is included in the NCOALink Processing Acknowledgement Form (PAF).
<i>Licensee Name</i>	This field is required for NCOALink. The name of the company, up to 30 characters, as mentioned in the license agreement with the USPS. The licensee performs the NCOALink processing. This information appears in the PAF log and NCOALink Processing Summary report.
<i>List Owner NAICS Code</i>	This is a required field. Enter the North American Industry Classification System (NAICS) code, which identifies the business in which the list owner engages. The provided substitution parameter for this option is \$\$CompanyNAICSCode. For more information, visit the NAICS Web site at http://www.census.gov/epcd/www/naics.html  .
<i>List ID</i>	The Customer or List ID is required for NCOALink limited and full service providers. End users may leave it blank. A unique ID assigned by the licensee to identify the list owner (customer). If the licensee does not have a naming scheme in place for the customer or lists, the six digits could be made up of the following: <ul style="list-style-type: none"> • First 3 digits: Customer name/identifier • Last 3 digits: List name identifier

Option	Description
<i>List Processing Frequency</i>	<p>This 2-digit number (from 1 to 52) indicates how many times per year the list is processed with NCOALink.</p> <p>If the list owner has other lists processed by the NCOALink licensee at different frequencies, enter 99.</p>
<i>List Received Date</i>	Enter the date when the NCOALink licensee received the list. Use the <code>yyyy/mm/dd</code> format. If you are an end user, you may leave this blank.
<i>List Return Date</i>	Enter the date when the list will be returned to the customer. Use the <code>yyyy/mm/dd</code> format. If you are an NCOALink end user, you may leave this blank.
<i>Provider Level</i>	<p>This option lists the provider levels for which you have a registered license keycode. It defaults to the substitution parameter <code>\$\$USPSProviderLevel</code>.</p> <p>Only provider levels supported in your registered keycodes display in the option list.</p>

8.5.12.6.1 Required options for USPS License Information

If you are processing NCOALink, DSF2, DPV, SuiteLink, or LACSLink, the USPS License Information group contains options that must be completed.

→ Tip

These options have substitution variables that you can set up in [Tools > Substitution Parameter Configurations](#).

Option	NCOALink Full Service Provider*	NCOALink Limited Service Provider ¹	NCOALink End User*	DSF2	DPV	LACSLink
Licensee Name	X	X	X	X		
List Owner NAICS Code	X	X	X	X		
List ID	X	X	X	X		
Customer Company Name	X	X	X	X	X	X
Customer Company Address	X	X	X	X	X	X
Customer Company Locality	X	X	X	X	X	X
Customer Company Region	X	X	X	X	X	X
Customer Company Post-code1	X	X	X	X	X	X
Customer Company Post-code2	X	X	X	X	X	X
Customer Company Phone						

¹ With NCOALink enabled

Option	NCOALink Full Service Provider*	NCOALink Limited Service Provider ¹	NCOALink End User*	DSF2	DPV	LACSLink
List Processing Frequency	X	X	X			
List Received Date	X	X		X		
List Return Date	X	X		X		
Provider Level	X	X	X			
IMB Mailer ID						
DSF2 Licensee ID				X		

8.5.12.7 NCOALink options

This section describes the options in the NCOALink group. The related links list the sub option groups.

Option	Description
Mailing List Name	Enter the name of this list, up to 30 characters. If this list is a master house list or your only mailing list, consider entering your company name here. This name appears in the log files.
Platform ID	The platform ID is the NCOALink licensee's identification number that is assigned by the USPS. It's exactly four characters long.

Related Information

[Processing options \[page 719\]](#)

[Report Options \[page 721\]](#)

[NCOALink Output Options \[page 722\]](#)

[Processing Acknowledgment Form \(PAF\) Details \[page 722\]](#)

[Service Provider Options \[page 723\]](#)

[Contact Detail List \[page 725\]](#)

8.5.12.7.1 Processing options

The following table describes the [NCOALink Processing Options](#).

¹ With NCOALink enabled

Option	Description
<i>Consider Moves Within Months</i>	<p>Use this setting to ignore change-of-address data older than the specified number of months. For example, enter 12 to use change-of-address data that has a move-effective date within the last 12 months.</p> <p>If you are an end user or limited service provider, enter a value from 6 to 18. If you're a full service provider or using ANKLink, enter a value from 6 to 48. If the option is blank, the transform uses all available data based on your license. The default is blank.</p>
<i>External Processes Updating List</i>	<p>Indicate whether the list undergoes additional processing before or after the USA Regulatory Address Cleanse transform.</p>
<i>High Match Rate Expectancy</i>	<p>The USPS wants to distinguish between files that have a legitimate reason for a high percentage of NCOALink matches and files that are fraudulently used to create mover lists. Select <i>None</i> or leave blank if you don't expect a high match rate. This option provides legitimate reasons for a high match rate.</p> <p><i>None</i>: Default.</p> <p><i>ANKLink Processed List</i>: An ANKLink-processed file contains records for people who have moved but you don't yet have their new address. This option is available only to full service providers.</p> <p><i>Stage File</i>: If you're performing Stage I or Stage II testing, ensure that the <i>List Processing Objective</i> is set to a Stage option.</p> <p><i>Return Mail List</i>: A returned mail list file contains records for mail that was returned to sender.</p>
<i>List Processing Mode</i>	<p><i>Change of Address</i>: You're processing this job to update it with the latest address data. Default.</p> <p><i>Statistics Only</i>: You're processing this job to analyze statistics such as the number of records in your list that have updated addresses and the number of moves of each type. When you choose this option, you do not receive move-updated addresses.</p> <p><i>Return Codes Only</i>: You're processing this job for informational purposes. When you choose this option and post to the NCOALink_Return_Code or ANKLink_Return_Code output component, you can see the return codes, which further explain whether matching records were found in the NCOALink directories and why or why not. With this option, you do not receive move-updated addresses.</p>

Option	Description
<i>List Processing Objective</i>	<p>Specify your reason for using NCOALink:</p> <p><i>Employee Training</i>: You're processing this file as part of employee training.</p> <p><i>Internal Database Testing</i>: You're testing with a licensee-owned database.</p> <p><i>Marketing</i>: You're testing with external customer lists.</p> <p><i>Normal</i>: You're processing the mailing list to update it before a mailing. Default.</p> <p><i>Stage I</i> and <i>Stage II</i>: You're testing the matching performance against a USPS test file. The USPS scores the Stage II test file. Choose Stage I or Stage II only if you are processing a USPS test file.</p> <p><i>System Testing</i>: You're processing this file as part of system testing such as loading USPS file updates.</p> <div> <p>i Note</p> <p>When certifying for CASS and DSF2, you indicate the reason in the Assignment Options > USPS Certification Testing Mode option.</p> </div>
<i>Processing First Class Mail</i>	Select the types of mail to process by selecting <i>Yes</i> or <i>No</i> for each option.
<i>Processing Periodicals</i>	<i>Processing First Class Mail</i> defaults to <i>Yes</i> ; the others default to <i>No</i> .
<i>Processing Standard Mail</i>	
<i>Processing Package Service Mail</i>	
<i>Retrieve Move Types</i>	<p>Choose the types of moves to process:</p> <p><i>Business</i>: Business moves only.</p> <p><i>Individual</i>: Individual moves only.</p> <p><i>Individual and Business</i></p> <p><i>Individual and Family</i></p> <p><i>Individual and Family and Business</i>: Default.</p>

8.5.12.7.2 Report Options

There is one option in the *NCOALink Report Options* group.

Option	Description
<i>Generate Return Code Descriptions</i>	<p>The NCOALink Processing Summary report always includes a brief summary of return codes, and you can include more detailed return code descriptions using this option. Return codes indicate whether a record was affected by a move, how the NCOALink match was made, or why a match could not be made.</p> <p>Yes: Includes the report codes in the NCOALink Processing Summary report.</p> <p>No: Default. Excludes report codes from the NCOALink Processing Summary report.</p>

8.5.12.7.3 NCOALink Output Options

There is one option in the *NCOALink Output Options* group.

Option	Description
<i>Apply Move to Standardized Fields</i>	<p>Component output fields are not affected by this option.</p> <p>Yes: Default. Data Services updates standardized fields to contain details about the address available through NCOALink.</p> <p>No: Standardized output fields have the standardized version of input rather than the moved address.</p>

8.5.12.7.4 Processing Acknowledgment Form (PAF) Details

The following table describes the NCOALink PAF Details. PAF Details are not required for end users.

Option	Description
<i>Company Website</i>	Enter the company website address for the person signing the PAF. You can leave this parameter blank.
<i>Customer Alternate Company Name</i>	If the list owner's company is also known by another name, enter that alternate name here.
<i>Customer Parent Company Name</i>	If the list owner's company is owned by another company, enter the parent company's name here.
<i>Date Signed By Licensee</i>	Enter the date that the licensee (the NCOALink service provider) signed the PAF in yyyy/mm/dd format.
<i>Date Signed By Customer</i>	Enter the date the customer signed the PAF in yyyy/mm/dd format.
<i>Email of Person Signing</i>	Enter the email address for the person who is signing the PAF. You can leave this parameter blank.
<i>Name Of Person Signing</i>	Enter the name of the person signing this PAF, up to 50 characters.

Option	Description
<i>Title Of Person Signing</i>	Enter the job title of the person signing this PAF, up to 50 characters.
<i>Type</i>	<p><i>Initial:</i> This is the first PAF you're completing to become authorized to process addresses for this particular customer.</p> <p><i>Modified:</i> You're completing a new PAF because some information on your old one changed.</p> <p><i>Renewal:</i> You're completing a new PAF because your old one is expiring.</p>
<i>Using Alternative PAF</i>	<p><i>Yes:</i> Select if you are using a PAF that is not the USPS form (you must have permission from the USPS).</p> <p><i>No:</i> The default setting for this field.</p> <p>This field requires either a <i>Yes</i> or <i>No</i>.</p>
<i>Using Cooperative Database</i>	<p>Indicates whether the list is from a cooperative database. Applicable for Full and Limited Service Providers only.</p> <p><i>Yes</i></p> <p><i>No:</i> (the default setting)</p> <div> <p>i Note</p> <p>A PAF must be on file for each participant in the cooperative database.</p> </div> <div> <p>i Note</p> <p>When set to Yes, a "C" is included in the PAF log to indicate that the list processed was a cooperative database.</p> </div>

8.5.12.7.5 Service Provider Options

The following table describes the *NCOALink Service Provider Options*. These options are not required for end users.

Option	Description
<i>Additional Notes</i>	<p><i>None:</i> Default.</p> <p><i>Customer Requested Extension:</i> Select if the customer submitted a written request for an extension.</p>
<i>Buyer Company Name</i>	If the list was processed for rent, sale, or lease, enter the name of the company or individual who bought the list.

Option	Description
<i>Concurrent Processed Data Modified</i>	<p><i>No</i>: If you are processing this data in some other way while performing NCOALink processing, indicates that the concurrent processing does not include changes to the data. Default.</p> <p><i>From Postal Data</i>: If you are processing this data in some other way while performing NCOALink processing, indicates whether the processing includes changes with postal data.</p> <p><i>From Non Postal Data</i>: If you are processing this data in some other way while performing NCOALink processing, indicates whether the processing includes changes with non-postal data.</p> <p><i>From Both</i>: If you are processing this data in some other way while performing NCOALink processing, indicates whether the processing includes changes with both postal and non-postal data.</p>
<i>Concurrent Processes Performed</i>	<p>Indicates whether you processed or will process this data in some other way while performing NCOALink processing.</p> <p><i>Yes</i></p> <p><i>No</i></p>
<i>In House List Processing</i>	<p>Indicates whether the list is an in-house (internal) list. Applicable for Full Service Providers only.</p> <p><i>Yes</i></p> <p><i>No</i></p> <div> <p>i Note</p> <p>When set to Yes, an "I" is included in the Customer Service log to indicate that the list was an in-house list.</p> </div>
<i>Output Returned</i>	<p>Identifies the type of output returned to the client.</p> <p><i>Standard</i>: All required NCOALink output was returned to the client. Default.</p> <p><i>Modified</i>: One or more post processes modified the return information (updates were applied to the list).</p> <p><i>Both</i>: One or more post processes modified the return information (updates were applied to the list); however, a separate file containing all of the required output was also returned.</p>

Option	Description
<i>Post Processed Data Modified</i>	<p><i>No</i>: If you are processing this data after performing NCOALink processing, indicates that the postprocessing does not include changes to the data. Default.</p> <p><i>From Postal Data</i>: If you are processing this data after performing NCOALink processing, indicates whether the postprocessing includes changes with postal data.</p> <p><i>From Non Postal Data</i>: If you are processing this data after performing NCOALink processing, indicates whether the postprocessing includes changes with non-postal data.</p> <p><i>From Both</i>: If you are processing this data after performing NCOALink processing, indicates that the postprocessing includes changes with both postal and non-postal data.</p>
<i>Post Processes Performed</i>	<p>Indicates whether you are processing this data after performing NCOALink processing.</p> <p><i>Yes</i></p> <p><i>No</i></p>
<i>Postcode For Mail Entry</i>	Specifies the ZIP Code of the Business Mail Entry Unit (BMEU) or post office where the mail will be submitted for mailing.
<i>Pre Processed Data Modified</i>	<p><i>No</i>: If you are processing this data before performing NCOALink processing, indicates that the preprocessing does not include changes to the data. Default.</p> <p><i>From Postal Data</i>: If you are processing this data before performing NCOALink processing, indicates whether the preprocessing includes changes with postal data.</p> <p><i>From Non Postal Data</i>: If you are processing this data before performing NCOALink processing, indicates whether the preprocessing includes changes with non-postal data.</p> <p><i>From Both</i>: If you are processing this data before performing NCOALink processing, indicates that the preprocessing includes changes with both postal and non-postal data.</p>
<i>Pre Processes Performed</i>	<p>Indicates whether you processed or will process this data before performing NCOALink processing.</p> <p><i>Yes</i></p> <p><i>No</i></p>

8.5.12.7.6 Contact Detail List

The following table describes the *NCOALink Contact Details* options that are located in the Contact Detail List group. These options are not required for end users.

Option	Description
Address	Enter the broker's or list administrator's address.
Contact Level	<p>Enter the degree of separation this contact is from you from 1 to 99. For example, enter 1 if you received the list from this contact. If your contact received the list from a different broker, then enter 2 for this contact.</p> <p>Note that the transform doesn't use this value in any logs.</p>
Contact Company Website	Enter the website of the broker or list administrator. You can leave this parameter blank.
Locality	Enter the broker's or list administrator's locality (city).
Type	<p>Broker: A broker directs business to the service provider.</p> <p>List Administrator: A list administrator stores and maintains address lists.</p> <p>Owner: The owner of the list processes it with no involvement of a broker or list administrator.</p>
License Assigned ID	Enter a unique six-character ID number for the broker or list administrator. You assign the ID number.
NAICS Code	Enter the broker's or list administrator's numeric North American Industry Classification System code, which identifies the business in which they engage. For more information, see http://www.census.gov/epcd/www/naics.html .
Name	Enter the broker's or list administrator's name.
PAF Signing Date	Enter the date when the contact signed the PAF in the format yyyy/mm/dd.
Phone	Enter the broker's or list administrator's phone number.
Postcode1	Enter the broker's or list administrator's Postcode1 (ZIP Code).
Postcode2	Enter the broker's or list administrator's Postcode2 (ZIP+4 Code).
Region	Enter the broker's or list administrator's region (state).

8.5.12.8 Assignment options

With this option group, you can choose the add-on features that you want to use during processing.

Option	Description
Dual Address	<p>Specifies the action to take when the transform encounters a dual address.</p> <p>Position: Selects an address based on the arrangement of the input data. The transform attempts to validate the address that is closest to the lower left corner of the address block. That might be the postal address (rural route or PO Box) or the street address; it depends on how the data was entered.</p> <p>Postal: The transform attempts to validate based on the postal address. If that fails, the transform attempts again based on the street address.</p> <p>Street: The transform attempts to validate based on the street address. If that fails, the transform attempts again based on the postal address.</p>

Option	Description
<i>Enable DPV</i>	<p>Specifies whether to perform DPV (Delivery Point Validation) processing.</p> <p><i>Yes:</i> Enables DPV processing.</p> <p><i>No:</i> Disables DPV processing.</p>
<i>Enable DSF2 Augment</i>	<p>Specifies whether to perform DSF2 (Delivery Sequence File Second Generation) augment processing.</p> <p><i>Yes:</i> Enables DSF2 augment processing.</p> <p><i>No:</i> Disables DSF2 augment processing.</p>
<i>Enable eLOT</i>	<p>Specifies whether to perform eLOT (Enhanced Line of Travel) processing.</p> <p><i>Yes:</i> Enables eLOT processing.</p> <p><i>No:</i> Disables eLOT processing.</p>
<i>Enable EWS</i>	<p>Specifies whether to perform EWS (Early Warning System) processing.</p> <p>If this transform cannot make an exact match within the <code>zip4us.dir</code> (Address Directory 1), it searches the EWS directory to see if the address is a new delivery point. If the address is located in the EWS directory, the transform marks the record as an EWS match and does not attempt further assignment.</p> <p><i>Yes:</i> Enables EWS processing.</p> <p><i>No:</i> Disables EWS processing.</p>
<i>Enable LACSLink</i>	<p>Specifies whether to perform LACSLink (Locatable Address Conversion System) processing.</p> <p><i>Yes:</i> Enables LACSLink processing.</p> <p><i>No:</i> Disables LACSLink processing.</p>
<i>Enable NCOALink</i>	<p>Specifies whether to perform NCOALink (National Change of Address) processing.</p> <p><i>Yes:</i> Enables NCOALink processing.</p> <p><i>No:</i> Disables NCOALink processing.</p>
<i>Enable RDI</i>	<p>Specifies whether to perform RDI (Residential Delivery Indicator) processing.</p> <p><i>Yes:</i> Enables RDI processing.</p> <p><i>No:</i> Disables RDI processing.</p>
<i>Enable Reverse Soundex Search</i>	<p>Specifies whether to use the <code>zip4us.rev</code> (Reverse Soundex) directory in an attempt to make address assignments.</p> <p><i>Yes:</i> Enables Reverse Soundex.</p> <p><i>No:</i> Disables Reverse Soundex.</p>
<i>Enable SuiteLink</i>	<p>Specifies whether to perform SuiteLink processing.</p> <p><i>Yes:</i> Enables SuiteLink processing.</p> <p><i>No:</i> Disables SuiteLink processing.</p>

Option	Description
<i>Geo Mode</i>	<p>Specifies which type of GeoCensus processing you want to perform.</p> <p><i>Address:</i> The transform processes address-level GeoCensus only.</p> <p><i>Both:</i> The transform makes an address-level GeoCensus assignment first if applicable. If no assignment is made, the transform makes a centroid-level GeoCensus assignment if applicable.</p> <p><i>Centroid:</i> The transform processes centroid-level GeoCensus only.</p> <p><i>None:</i> The transform turns off GeoCensus processing. Choose this option if you have not purchased the GeoCensus option or if you do not want to perform GeoCensus processing.</p> <div> <p>i Note</p> <p>GeoCensus functionality in the USA Regulatory Address Cleanse transform will be deprecated in a future version. It is recommended that you upgrade any data flows that currently use the GeoCensus functionality to use the Geocoder transform. For instructions on upgrading from GeoCensus to the Geocoder transform, see the <i>Upgrade Guide</i>.</p> </div>
<i>USPS Certification Testing Mode</i>	<p>Indicates the type of certification being processed so that the software checks for the appropriate settings and issues warnings and errors when applicable.</p> <p><i>None:</i> The transform processes the job normally, without any special settings for certification. This is the default setting.</p> <p><i>CASS:</i> The transform processes the job with the appropriate settings for CASS self-certification.</p> <p><i>DSF2 Augment:</i> The transform processes the job with appropriate settings for DSF2 augment certification.</p> <div> <p>i Note</p> <p>When certifying for NCOALink, you set the testing mode in the ► NCOALink ► Processing Options ► List Processing Objective ► option.</p> </div>

8.5.12.9 Standardization options

This option group contains all of the standardization settings that you need to define for processing USA data. (The options are listed alphabetically.)

Option	Description
<i>Add Firm Match Secondary</i>	<p><i>Yes</i>: Adds secondary address information obtained from SuiteLink directories to the address line.</p> <p><i>No</i>: Does not add secondary address information obtained from SuiteLink directories to the address line, but includes SuiteLink-found information reflected in the lastline ZIP+4 Code and in other output fields.</p> <p>CASS users who do not want to update address lines in their data with SuiteLink secondary information should set this option to No. The software updates the last line to reflect the SuiteLink secondary information in the ZIP+4, and does not update the original address. The software also updates the address line based on your standardization settings in the job setup.</p> <p>Sometimes the option setting has no effect on the presence of the SuiteLink secondary address information. This can happen when the secondary address does not match the National directories, but exactly matches the SuiteLink secondary address information. In this situation, the software does not consider the SuiteLink secondary address as a change to the input secondary address, and does not remove the secondary address from the output address even when the Add Firm Match Secondary option is set to No. This applies to an exact match to the SuiteLink secondary unit and range, or, in cases where there is no unit designator on input, an exact match to the SuiteLink secondary range.</p>
<i>Address Line Alias</i>	<p>Specifies how to standardize the address line if the input primary address is an alias.</p> <p><i>Convert</i>: Converts address lines to the preferred form found in the postal directory.</p> <p><i>Preserve</i>: Retains address lines as they were input.</p> <div> <p>i Note</p> <p>To be compliant with CASS, set up your jobs to return the USPS preferred address. When the <i>Address Line Alias</i> option is set to <i>Convert</i>, the USPS preferred address is returned, even when the input record has a base address or an alias address. You can choose to set up your job to preserve the preferred address (<i>Address Line Alias</i> set to <i>Preserve</i>), but the software does not produce a USPS 3553 form.</p> </div>
<i>Append Private Mailbox</i>	<p>Private mailboxes (PMB) are like post office boxes, except that they are hosted by private companies.</p> <p><i>Yes</i>: Places the address and PMB in the same field.</p> <p><i>No</i>: Places the PMB into a separate field. PMB information is output to the NON_POSTAL_SECONDARY_ADDRESS, NON_POSTAL_UNIT, and NON_POSTAL_UNIT_NUMBER fields.</p>
<i>Assign With Input Locality</i>	<p>Specifies whether to use the last-line index when assigning the locality (city) name.</p> <p><i>Yes</i>: Assigns the Locality1 based on the locality name that is input if it is valid for the Postcode1. Does not change the Locality1 based on last line index.</p> <p><i>No</i>: Assigns the Locality1 based on the locality that is input if it is valid for the Postcode1 and not a place name; otherwise, assigns Locality1 based on the last-line index of the address line. Produces a more geographically true Locality1. If you choose <i>No</i>, the value you choose for the <i>Preserve Place Names</i> option does not matter; place names are converted.</p> <div> <p>i Note</p> <p>When the <i>Use USPS Locality Abbreviation</i> option is set to <i>Yes</i>, <i>Preserve Place Names</i> is set to <i>Yes</i>, and <i>Assign with Input Locality</i> is set to <i>No</i>, the software may not preserve some place names over 13 characters and abbreviates them.</p> </div>

Option	Description
<i>Capitalization</i>	<p>Specifies the casing of your address data.</p> <p><i>Lower:</i> Converts data to all lowercase letters. For example, "Main Street South" becomes "main street south."</p> <p><i>Mixed:</i> Converts data to initial capital letters. For example, "MAIN STREET SOUTH" becomes "Main Street South."</p> <p><i>Upper:</i> Converts data to all capital letters. For example, "Main Street South" becomes "MAIN STREET SOUTH."</p> <div> <p>i Note</p> <p>If you want consistent casing for your data, make sure that this option and the <i>Capitalization</i> setting in the Data Cleanse transform are the same.</p> </div>
<i>Combine Multilines</i>	<p>Specifies what to do with related fields input on separate lines.</p> <p><i>Yes:</i> Looks for related fields that were input on separate lines, and tries to put them together on the same line.</p> <p><i>No:</i> Does not try to combine fields.</p>
<i>Directional Style</i>	<p>Specifies whether to abbreviate directional data.</p> <p><i>Long:</i> Uses fully-spelled directionals such as "North," "South," "East," and "West."</p> <p><i>Preserve:</i> Preserves the style used in the input record.</p> <p><i>Short:</i> Uses abbreviated directionals such as "N," "S," "E," and "W."</p> <div> <p>i Note</p> <p>When the <i>Use USPS Street Abbreviation</i> option is set to <i>Yes</i>, the software overrides a setting of <i>Long</i> and <i>Preserve</i> for the <i>Directional Style</i> option and outputs the short directional style.</p> </div>

Option	Description																																	
<i>Include Unused Address Line Data</i>	<p>Specifies whether to output the unused address line data (for discrete and multiline fields).</p> <p><i>Yes:</i> Includes the unused address line data, including invalid secondary information, in the address line and in the ADDRESS_LINE_REMAINDER1 field.</p> <p><i>No:</i> Does not output the unused address line data in the address line but outputs the information to certain output fields.</p> <p>Example:</p> <table><tr><th>Input</th><th colspan="2">Include Unused Address Line Data option</th></tr><tr><th></th><th>Yes</th><th>No</th></tr><tr><td>SAP</td><td>SAP</td><td>SAP</td></tr><tr><td>332 FRONT ST S FL 9</td><td>332 FRONT ST S FL 3 FL 9</td><td>332 FRONT ST S FL 3</td></tr><tr><td>LA CROSSE WI 54601</td><td>LA CROSSE WI 54601-4025</td><td>LA CROSSE WI 54601-4025</td></tr></table> <p><i>Output fields</i></p> <table><tr><td>Primary_Secondary_Address</td><td>332 FRONT ST S FL 3</td><td>332 FRONT ST S FL 3</td></tr><tr><td>Full_Address</td><td>332 FRONT ST S FL 3 FL 9</td><td>332 FRONT ST S FL 3</td></tr><tr><td>Address_Line_Remainder1</td><td>FL 9</td><td>FL 9</td></tr><tr><td>Secondary_Address</td><td>FL 3</td><td>FL 3</td></tr><tr><td>Pre_SuiteLink_Unit_Description</td><td>FL</td><td>FL</td></tr><tr><td>Pre_SuiteLink_Unit_Number</td><td>9</td><td>9</td></tr></table>	Input	Include Unused Address Line Data option			Yes	No	SAP	SAP	SAP	332 FRONT ST S FL 9	332 FRONT ST S FL 3 FL 9	332 FRONT ST S FL 3	LA CROSSE WI 54601	LA CROSSE WI 54601-4025	LA CROSSE WI 54601-4025	Primary_Secondary_Address	332 FRONT ST S FL 3	332 FRONT ST S FL 3	Full_Address	332 FRONT ST S FL 3 FL 9	332 FRONT ST S FL 3	Address_Line_Remainder1	FL 9	FL 9	Secondary_Address	FL 3	FL 3	Pre_SuiteLink_Unit_Description	FL	FL	Pre_SuiteLink_Unit_Number	9	9
Input	Include Unused Address Line Data option																																	
	Yes	No																																
SAP	SAP	SAP																																
332 FRONT ST S FL 9	332 FRONT ST S FL 3 FL 9	332 FRONT ST S FL 3																																
LA CROSSE WI 54601	LA CROSSE WI 54601-4025	LA CROSSE WI 54601-4025																																
Primary_Secondary_Address	332 FRONT ST S FL 3	332 FRONT ST S FL 3																																
Full_Address	332 FRONT ST S FL 3 FL 9	332 FRONT ST S FL 3																																
Address_Line_Remainder1	FL 9	FL 9																																
Secondary_Address	FL 3	FL 3																																
Pre_SuiteLink_Unit_Description	FL	FL																																
Pre_SuiteLink_Unit_Number	9	9																																

Option	Description
Move Multiline Data	<p>If you turn on this feature, the USA Regulatory Address Cleanse transform rearranges your multiline data to conform to USPS guidelines.</p> <p>The transform moves the primary address into position above the locality-region-postal code line (or lastline).</p> <p>Bottom: Rearranges the lines according to USPS guidelines. If there are any blank lines, the transform moves them to the top and shifts the data to the bottom of the block.</p> <p>No: Does not rearrange any lines, blank or otherwise.</p> <p>Top: Rearranges the lines according to USPS guidelines. If there are any blank lines, the transform moves them to the bottom and shifts the data to the top of the block.</p> <p>Example:</p> <div data-bbox="391 799 1091 1086"> <p>Input data:</p> <p>Line1 100 Market Street</p> <p>Line2 Suite 202</p> <p>Line3 Sycamore Building</p> <p>Line4</p> <p>Line5 Boston MA 02109</p> <p>Result of moving:</p> <p>Sycamore Building</p> <p>Suite 202</p> <p>100 Market St</p> <p>Boston MA 02109</p> </div> <p>This feature does not require that you standardize your data.</p> <p>Note: If you choose Top or Bottom, the input field lengths of all fields mapped to Multiline<x> must be the same. For example, if Multiline1 is set to a length of 60, all Multiline fields that you use must be set to 60.</p>
Multiline Update Postcode 1	<p>This option affects multiline data only (data passed in and retrieved through multiline fields).</p> <p>Dont_Update: Assigns Postcode1 fields, but does not write them to the Multiline output fields. In those fields, leaves the original Postcode1 intact and the assigned Postcode1 is available in other output fields.</p> <p>Erase_Then_Update: Replaces the original Postcode1 with the assigned Postcode1 in the Multiline output fields. If no Postcode1 is assigned, the original Postcode1 is not included in the Multiline output fields.</p> <p>Update: Replaces the input Postcode1 with the assigned Postcode1 in the Multiline output fields. If no Postcode1 is assigned, the original is retained.</p>

Option	Description
<i>Multiline Update Postcode 2</i>	<p>This option affects multiline data only (data passed in and retrieved through multiline fields).</p> <p><i>Dont_Update</i>: Assigns Postcode2, but does not write them to the Multiline output fields. The transform leaves the original Postcode2 intact and the assigned Postcode2 is available in other output fields.</p> <p><i>Erase_Then_Update</i>: In the Multiline output fields, replaces the original Postcode2 with the assigned Postcode2. If no Postcode2 is assigned, the original Postcode2 is not available in the Multiline output fields.</p> <p><i>Update</i>: In the Multiline output fields, replaces the input Postcode2 with the assigned Postcode2. If no Postcode2 is assigned, it retains the original.</p>
<i>Preserve Dual Address Order</i>	<p>Specifies whether to preserve or change the dual address order.</p> <p><i>Yes</i>: When an address contains both a street and mailing address, keeps the address order as it was input.</p> <p><i>No</i>: When the input address contains both a locality and mailing address, moves the assigned address immediately above the locality and region.</p>
<i>Preserve Place Names</i>	<p>Specifies whether to preserve or change non-mailing city names (place names).</p> <p><i>Yes</i>: Preserves the non-mailing city name. Given Hollywood as input, the transform outputs Hollywood.</p> <p><i>No</i>: Changes non-mailing city names to city names preferred by the USPS. Given Hollywood as input, the transform outputs Los Angeles.</p> <div> <p>i Note</p> <p>When the <i>Use USPS Locality Abbreviation</i> option is set to <i>Yes</i>, <i>Preserve Place Names</i> is set to <i>Yes</i>, and <i>Assign with Input Locality</i> is set to <i>No</i>, the software may not preserve some place names over 13 characters and abbreviates them.</p> </div>
<i>Primary Type Style</i>	<p>Specifies whether to abbreviate the street (primary) type.</p> <p><i>Long</i>:: Uses fully-spelled primary types such as Street, Avenue, and Road.</p> <p><i>Preserve</i>: Preserves the style used in the input record.</p> <p><i>Short</i>: Uses abbreviated primary types such as St, Ave, and Rd.</p> <div> <p>i Note</p> <p>When <i>Use USPS Street Abbreviation</i> is set to <i>Yes</i>, the software overrides a setting of <i>Long</i> and <i>Preserve</i> for the <i>Primary Type Style</i> option and outputs the short primary type style.</p> </div>

Option	Description
<i>Retain Pound Sign in Unit Description</i>	<p>Outputs “#” into either extraneous fields or to the UNIT_DESCRIPTION output field.</p> <p><i>Yes:</i> Outputs the # unit designator to the UNIT_DESCRIPTION output field.</p> <p><i>No:</i> Outputs the # unit designator to the EXTRANEOUS_SECONDARY_UNIT_NUMBER and/or the EXTRANEOUS_SECONDARY_ADDRESS_DATA output fields.</p> <p>The option does not affect the following address situations:</p> <ul style="list-style-type: none"> • Puerto Rican addresses • Military addresses • Rural Route addresses • Addresses without “#” in the address line • Addresses with remainder words
<i>Standardize Assigned Address</i>	<p>Specifies whether to correct and standardize the assigned address line and lastline data.</p> <p><i>Yes:</i> Corrects and standardizes your address line and lastline data. Use this value for CASS certification.</p> <p><i>No:</i> Does not standardize your address line or lastline data.</p>
<i>Standardize Unassigned Address</i>	<p>Specifies whether to standardize unassigned data.</p> <p><i>Yes:</i> Attempts to parse and standardize any unassigned addresses.</p> <p><i>No:</i> Leaves unassigned addresses as entered on input.</p>
<i>Unit Description</i>	<p>Specifies how to standardize the unit description.</p> <p><i>Convert:</i> Uses the unit description found in the postal directory (such as an apartment, suite, room, or floor).</p> <p><i>Preserve:</i> Preserves the unit description from the input record and corrects any spelling errors.</p>

Option	Description
<i>Use USPS Locality Abbreviation</i>	<p>Yes: Enables this option (and affects only the multiline and standardized last line fields).</p> <p>No: Disables this option.</p> <p>Provides a USPS 13-character city name when one is available. If the city name is not valid (for example, it is a non mailing city name), the software relies on other settings in the job to determine what to output for city.</p> <p>If the city name is longer than 13 characters, the software returns an abbreviation that is 13 characters or less. If the city name is already 13 characters or less, the software does not abbreviate it.</p> <div> <p>i Note</p> <p>When the number of characters in the output is greater than the length specified for the output field, the software attempts to truncate the output data to fit in the output field without eliminating vital address data.</p> <p>Intelligent truncation abbreviates the output data first, and if it still doesn't fit the output buffer, it truncates the data.</p> <p>There are no options to set this up in the USA Regulatory Address Cleanse transform. The transform does this automatically.</p> <p>If the <i>Use USPS Primary Name Abbreviation</i> and/or the <i>Use USPS Locality Abbreviation</i> options are enabled, the software uses those abbreviations first. If the values don't fit within the length of the output fields, then intelligent truncation occurs.</p> </div>

Option	Description
<i>Use USPS Primary Name Abbreviation</i>	<p><i>Yes</i>: Enables this option.</p> <p><i>No</i>: Disables this option.</p> <p>Abbreviates the address line to 30 characters or less when the output address line exceeds 30 characters and when an abbreviated form of the address is available in the directory data supplied by the USPS. Abbreviated forms of an address are only provided for output addresses 30 characters or greater. If the output address line is already 30 characters or less, the output address line is not abbreviated.</p> <div> <p>i Note</p> <p>An address line may be output with more than 30 characters in situations where no abbreviated form of the address is available in the directory data. If your data must fit exactly into 30 characters, we recommend to set appropriate address output field lengths to 30.</p> </div> <p>This option affects multiline and standardized last line fields when set to <i>Yes</i>.</p> <p>If the <i>Use USPS Street Abbreviation</i> option is set to <i>Yes</i>, it affects the following address components on output:</p> <ul style="list-style-type: none"> • <i>Suffix Style</i>: The style will be short. • <i>Directional Style</i>: The style will be short. • <i>Address Line Alias</i>: The setting of <i>Preserve</i> may be overridden. <div> <p>i Note</p> <p>When the number of characters in the output is greater than the length specified for the output field, the software attempts to truncate the output data to fit in the output field without eliminating vital address data.</p> <p>Intelligent truncation abbreviates the output data first, and if it still doesn't fit the output buffer, it truncates the data.</p> <p>There are no options to set this up in the USA Regulatory Address Cleanse transform. The transform does this automatically.</p> <p>If the <i>Use USPS Primary Name Abbreviation</i> and/or the <i>Use USPS Locality Abbreviation</i> options are enabled, the software uses those abbreviations first. If the values don't fit within the length of the output fields, then intelligent truncation occurs.</p> </div>

8.5.12.10 Non Certified options

This option group includes options to process your data without following CASS certification rules.

Option	Description
<i>Accept Inexact Postcode Move</i>	<p>When an input record has an obsolete postcode or a postcode move, specifies whether the transform should ignore some of the non-matching elements between the input record and the record in the national directory, and use built-in matching thresholds to determine if the records match.</p> <p>Yes: Ignores the non-matching elements and uses built-in matching thresholds.</p> <p>No: Disables this option (and does not ignore non-matching elements).</p>
<i>Assign With Input Postcode</i>	<p>Specifies whether the transform should use the last four digits of the 9-digit postcode (if present on input), which is usually the Postcode2 field (ZIP+4), during address assignment.</p> <p>Yes: Enables the transform to use the record's last four digits of the 9-digit postcode to try to make a finer level of assignment than it could make under CASS rules. Under CASS rules, the transform doesn't consider the last four digits of the 9-digit postcode. In order for this option to work, the last four digits of the 9-digit postcode must be unique to a valid firm or secondary address.</p> <p>No: Disables this option.</p>
<i>Assign Postcode2 Not DPV Validated</i>	<p>Specifies whether to output the Postcode2 when an assignment is made even when <i>Enable DPV</i> is set to No and <i>Disable Certification</i> is set to Yes. The output address is not validated by DPV.</p> <p>Yes: Assigns the Postcode2 when <i>Enable DPV</i> is set to No.</p> <p>No: Leaves the Postcode2 blank when an assignment is made and <i>Enable DPV</i> is set to No.</p>
<i>Disable Certification</i>	<p>Specifies whether to run address cleansing for CASS certification.</p> <p>Yes: Runs address cleansing without the restrictions of CASS certification rules. Choose this value if you want to use any of the other options in this option group. This value also enables non-mailers to process addresses for 14 months after the directory creation date rather than 3 to 4 months for postal discounts through the USPS. You will not receive any postal discounts with this value or be able to produce a USPS Form 3553.</p> <p>No: Runs address cleansing under CASS certification rules. The other options in this option group are ignored.</p>
<i>Enable Geo Only</i>	<div> <p>i Note</p> <p>GeoCensus functionality in the USA Regulatory Address Cleanse transform will be deprecated in a future version. It is recommended that you upgrade any data flows that currently use the GeoCensus functionality to use the Geocoder transform. For instructions on upgrading from GeoCensus to the Geocoder transform, see the <i>Upgrade Guide</i>.</p> </div> <p>Specifies whether the transform should process only with the address-level GeoCensus or centroid GeoCensus directories. Choose the type of GeoCensus processing in the Assignment Options group.</p> <p>Yes: Runs address cleansing with the GeoCensus directories only. The transform does not process your data with the postal directories. Make sure that you have defined the location of the GeoCensus directories in the Reference Files option group. You must also include the appropriate GeoCensus output fields; data is not posted in any other output fields.</p> <p>No: Enables you to run address cleansing with the GeoCensus directories, as well as the postal directories.</p>

Option	Description
<i>Enable Parse Only</i>	<p>Specifies whether the transform should parse and validate your data or parse only.</p> <p><i>Yes:</i> Parses records into their discrete components, but does not perform a lookup in the postal directories. This mode is fast, but parsing results are unverified.</p> <p><i>No:</i> Parses records into their discrete components and performs a lookup in the postal directories. This mode may be slower, but parsing results are verified.</p>
<i>Enable Suggestion Lists</i>	<p>Specifies whether suggestion lists are generated for records that cannot be assigned. This option is for transactional projects.</p> <p><i>Yes:</i> Generates suggestion lists.</p> <p><i>No:</i> Does not generate suggestion lists.</p>

8.5.12.11 CASS Report options

With this option group, you add the necessary USPS Form 3553 information as required by the USPS when certifying a mailing.

Option	Description
<i>Company Name Certified</i>	If you rely on SAP for vendor CASS certification, leave this parameter blank; the transform inserts "SAP" as the default value. If you have your own end-user CASS certification from the USPS, type your company name (up to 40 characters).
<i>List Name</i>	Specifies the name of the mailing list (up to 20 characters).
<i>List Owner</i>	Specifies the name of your company (up to 19 characters).
<i>LOT Certification</i>	<p>Specifies whether you have LOT certification.</p> <p><i>Yes:</i> You have LOT certification but you do not have CASS certification in your own name.</p> <p><i>No:</i> You have CASS certification in your own name but you did not seek or obtain LOT certification.</p> <p>In this case, setting LOT Certified to No ensures that the LOT Certification lines on your USPS 3553 forms are blank, which is appropriate.</p>
<i>Mailer Address 1</i>	Specifies the name and address of the person or organization for whom you are preparing the mailing (up to 29 characters per line).
<i>Mailer Address 2</i>	
<i>Mailer Address 3</i>	
<i>Mailer Address 4</i>	
<i>Software Version</i>	<p>If you rely on SAP for vendor CASS certification, you may leave this parameter blank. The transform inserts the appropriate software name and version as the default value.</p> <p>If you have received end-user CASS certification in your own company's name, type the software name and version number that you use to receive CASS certification.</p>

8.5.12.12 Suggestion List group

Set the options in this group to configure how suggestion lists are output.

Option/Option group	Description
<i>Address Lines Match Minimum</i>	<p>Specifies the similarity score required for address-line suggestions. Valid values are 0 to 80.</p> <p>The similarity score determines which suggestions are returned in the list. A higher number indicates that the suggestion must be more similar to the input to be returned as a possible suggestion.</p>
<i>Address Range Window</i>	<p>Specifies a number that represents a span. The software uses the number to present a range of addresses around the input primary address range for which to return suggestions.</p> <p>By using this option, you can limit the suggestions returned to be within a few blocks of your input. For example, assume you entered 500 for this value. Then, you submit the following street address:</p> <p>1000 Pine St.</p> <p>Suggestions would only be returned in a range from 750 to 1250 Pine Street.</p> <p>Type "0" if you don't want to limit the ranges returned in suggestions.</p>
<i>Combine Overlapping Ranges</i>	<p>Specifies how individual suggestions with overlapping ranges are consolidated.</p> <p><i>Combine_Ignoring_Gaps</i>: Ignores gaps and overlaps in primary ranges, so consolidation is more aggressive.</p> <p><i>Combine_Preserving_Gaps</i>: Preserves gaps in primary ranges, but overlapping ranges are consolidated.</p> <p><i>None</i>: Suggestions are not consolidated.</p>
<i>Delimiter</i>	<p>Specifies the delimiter to use between each suggestion. This is applicable if you chose <i>Delimited</i> for the <i>Style</i> option.</p> <p>Choose any character or string to separate each suggestion. This value should differ from the <i>Field Delimiter</i> value.</p>
<i>Field Delimiter</i>	<p>Specifies the delimiter to use between each suggestion list. This is applicable if you chose <i>Delimited</i> for the <i>Style</i> option.</p> <p>This value should differ from the <i>Delimiter</i> value.</p>
<i>Lastlines Match Minimum</i>	<p>Specifies the similarity score required for lastline suggestions.</p> <p>Enter a value from 0 to 80.</p> <p>The similarity score determines which suggestions are returned in the list. A higher number indicates that the suggestion must be more similar to the input to be returned as a possible suggestion.</p>

Option/Option group	Description
<i>Match Range</i>	<p>Specifies whether to disregard an address-line suggestion when it does not match the primary range of the input address.</p> <p><i>Yes:</i> Returns address-line suggestions only when they match the primary range of the input address.</p> <p><i>No:</i> Returns a possible address-line suggestion when it doesn't have the same primary range as the input.</p>
<i>Max Number Addresslines</i>	<p>Specifies the maximum number of address-line suggestions that can be generated. The maximum number that you can enter is 100.</p> <p>For example, you could set this option to limit the size of the SOAP documents being sent by the web service, or to limit the maximum number of suggestions that your users would have to choose from.</p> <div> <p>i Note</p> <p>If you set a low maximum, a viable suggestion could be left out of the suggestion list.</p> </div>
<i>Max Number Lastlines</i>	<p>Specifies the maximum number of lastline suggestions that can be generated. The maximum number that you can enter is 15.</p> <p>For example, you could set this option to limit the size of the SOAP documents being sent by the web service, or to limit the maximum number of suggestions that your users would have to choose from.</p> <div> <p>i Note</p> <p>If you set a low maximum, a viable suggestion could be left out of the suggestion list.</p> </div>
<i>Style</i>	<p>Specifies the style of the output file.</p> <p><i>Delimited:</i> Outputs the suggestion list data in a delimited text format, with the delimiters specified in the <i>Delimiter</i> and <i>Field Delimiter</i> options.</p> <p><i>XML:</i> Outputs the suggestion list data as hierarchical XML. This option is likely the preferred one for users who integrate suggestion lists via the web service. You can then use the XML tools that you own to parse the suggestion list data.</p>
<i>Suggestion List Components</i>	<p>Specifies the address field components that you want to include in the Suggestion_List output field.</p> <div> <p>i Note</p> <p>Suggestion list field components that do not have a value are not output to the Suggestion_List output field if the selected <i>Style</i> is <i>XML</i>.</p> </div>

Related Information

[Suggestion List components \[page 741\]](#)

8.5.12.13 Suggestion List components

These options let you choose what information to output to the Suggestion_List output field.

i Note

If XML is the selected Style in the Suggestion List option group, suggestion list fields that do not have a value are not output to the Suggestion_List output field.

Option/option group	Description
<i>Firm</i>	Select Yes to output the firm name for the secondary address.
<i>Locality1</i>	Select Yes to output the locality1 (city) preferred by the USPS. Applicable for primary, secondary, and lastline address levels.
<i>Postcode</i>	Select Yes to output the five-digit Postcode1 (not including the four-digit ZIP4). Applicable for primary, secondary, and lastline address levels.
<i>Postcode2 Even</i>	Select Yes to output the four-digit ZIP4 code, even numbers only. Applicable for primary and secondary address levels.
<i>Postcode2 Odd</i>	Select Yes to output the four-digit ZIP4 code, odd numbers only. Applicable for primary and secondary address levels.
<i>Primary Name1</i>	Select Yes to output the street name description. Applicable for primary and secondary address levels. <div>i Note Primary numbers (high or low) are not output to this field.</div>
<i>Primary Name Full1</i>	Select Yes to output the primary address line, such as the street address or post office box. The output includes the following information: Primary Prefix1, Primary Name1, Primary Type1, and Primary Postfix1. Applicable for primary and secondary address levels. <div>i Note Primary numbers (high or low) are not output to this field.</div>
<i>Primary Number High</i>	Select Yes to output the high portion of the premise number range. Applicable for primary and secondary address levels.
<i>Primary Number Low</i>	Select Yes to output the low portion of the premise number range. Applicable for primary and secondary address levels.
<i>Primary Postfix1</i>	Select Yes to output the abbreviated directional (for example, N, S, NW, or SE) that follows a street name. Applicable for primary and secondary address levels.
<i>Primary Prefix1</i>	Select Yes to output the abbreviated directional (for example, N, S, NW, or SE) that precedes a street name. Applicable for primary and secondary address levels.
<i>Primary Side Indicator</i>	Select Yes to output Odd or Even for the primary side indicator. Applicable for primary and secondary address levels.
<i>Primary Type1</i>	Select Yes to output the abbreviated street type (for example, St, Ave, or Pl). Applicable for primary and secondary address levels.

Option/option group	Description
<i>Region1</i>	Select Yes to output the state, province, territory, or region. Applicable for primary, secondary, and lastline address levels.
<i>Secondary Side Indicator</i>	Select Yes to output Odd or Even for the secondary side indicator. Applicable for secondary address level.
<i>Selection</i>	Select Yes to output the selection number for multiple suggestions.
<i>Unit Description</i>	Select Yes to output the unit description (for example, #, Apartment, or Flat). Applicable for secondary address level.
<i>Unit Number High</i>	Select Yes to output the high portion of the unit number range. Applicable for secondary address level.
<i>Unit Number Low</i>	Select Yes to output the low portion of the unit number range. Applicable for secondary address level.

8.5.12.14 Z4 Change options

With the Z4Change options you can turn on Z4Change processing and specify the last time the Postcode2 was updated.

Option	Description
<i>Enable Z4 Change</i>	Specifies whether to enable Z4Change processing. <i>Yes:</i> Turns on Z4Change processing. <i>No:</i> Turns off Z4Change processing.
<i>Last ZIP4 Assign Date</i>	Specifies the month and year that the input records were most recently ZIP+4 coded—either through a full address correction process or a previous Z4Change pass. Enter the date using the format MM/YYYY. For example, enter a date of January 2004 by typing 01/2004 . The USA Regulatory Address Cleanse transform verifies that your date is within the 12-month period covered by the Z4Change file. If there is a date problem, you will receive an error message when Data Quality runs your project.

8.5.12.15 USA Regulatory Address Cleanse fields

The USA Regulatory Address Cleanse transform requires that you map fields on input and output.

Related Information

[Input fields \[page 745\]](#)

8.5.12.15.1 Field category columns in Output tab

The Output tab lists output fields that hold the data that the transform cleanses or creates. You can choose to view the Best Practice, In Use, or All output fields by selecting the corresponding option at the top of the tab.

Best Practice: Lists all available output fields that have a field class of Best.

In Use: Lists only the output fields that you have chosen to output (listed in Schema Out).

All: Lists all output fields that are available for this transform.

Note

For details about mapping input and output fields, see the *Designer Guide*.

The output field attributes in the following table are listed in groups based on the field category column. Each field has categories that describe the type of content that is output. The field category displays “None” when it does not apply to the field.

Field category	Description
Content Type	Identifies the type of data in the field. Setting the content type helps you map your fields when you set downstream transforms. Content types Identifies the type of data in the field. Setting the content type helps you map your fields when you set downstream transforms.
Field Addrclass	<p>Specifies the address class for the generated field.</p> <p><i>Delivery</i>: When used with the applicable Field Name, this value generates fields that reflect the address that is used in an attempt to assign an address.</p> <p><i>Dual</i>: When used with the applicable Field Name, this value generates fields that reflect the address that is not used in an attempt to assign an address for input records that may contain both a street and postal address on input.</p> <p><i>Official</i>: When used with the applicable Field Name, this value generates fields in the form of the data preferred by the Postal Authority.</p> <p>For example, in Winona, Minnesota USA, Broadway and 6th Street are alternate names for the same street. A letter addressed to Broadway is delivered, but the USPS prefers 6th Street.</p>
Field Category	<p><i>Component</i>: Individual address components and postal codes that are related to the processed record.</p> <p><i>Standardized</i>: Standardized input lines based on the settings in the Standardization Options group in the transform.</p> <p><i>Suggestion</i>: Suggestion list output data based on the settings in the Suggestion List Options group.</p>

Field category	Description
Field Class (Delivery:USA Regulatory Address Cleanse)	<p>Specifies the field class that you want to assign to your output fields.</p> <p><i>Best</i>: Outputs data based on various factors, such as whether an address was assigned, the Field AddrClass, and any settings that you defined in the Standardization Options group in the Options tab.</p> <div> <p>i Note</p> <p>When NCOALink is enabled and a valid move is available, Best fields contain the move-updated address data if it exists and if it matches in the U.S. National Directories. Or, the field contains the original address data if a move does not exist or if the move does not match in the U.S. National Directories.</p> </div> <p><i>Correct</i>: Outputs the complete and correct value found in the directories, and is standardized according to any settings that you defined in the Standardization Options group in the Options tab.</p> <p><i>Parsed</i>: Outputs the parsed value.</p> <p><i>Pre_LACSLink</i>: Retained address components that were replaced with LACSLink address information.</p> <p><i>Move_Updated</i>: Outputs the address components that have been updated with move-updated address data.</p> <div> <p>i Note</p> <p>The transform looks for the move-updated address information in the U.S. National Directories. When the move-updated address is not found in the U.S. National Directories, the software populates the Move Updated fields with information found in the Move Update Directories only. The Move Updated fields that are populated as a result of standardizing against the U.S. National Directories is not updated.</p> </div>
Field Class (Global Address Cleanse)	<p>Specifies the field class that you want to assign to your output fields.</p> <p><i>Best</i>: Outputs data based on various factors, such as whether an address was assigned, the Field AddrClass, and any settings that you defined in the Standardization Options group in the Options tab.</p> <p><i>Parsed</i>: Outputs the parsed value.</p>
Field Name	Specifies a field name where the data is populated based on the options that you specify within this transform.
Type	Specifies the type and default length of data the output field contains; for example, varchar, date, and time.

Parent topic: [Global Address Cleanse fields \[page 597\]](#)

Related Information

[Input fields for the Global Address Cleanse transform \[page 601\]](#)

[NW input fields \[page 605\]](#)

[Mapping NW input fields \[page 607\]](#)

[Output fields for the Global Address Cleanse transform \[page 608\]](#)

[NW_PO_Box output fields \[page 631\]](#)

[Global Address Cleanse Suggestion List fields \[page 633\]](#)

[Content types \[page 445\]](#)

8.5.12.15.2 Input fields for the USA Regulatory Address Cleanse transform

The following are Data Services input fields that you can use to map the input data file fields for the USA Regulatory Address Cleanse transform.

Input field name (USA Regulatory Address Cleanse)	Description
Address_Line	The delivery address line (for example, "123 Main Street, Unit 4").
Check_Digit	<p>The check-digit for 11-digit delivery-point bar code. Applicable only if the transform can make a full assignment.</p> <p>The transform provides the check digit for a five-digit bar code when a five-digit assignment is possible, or the address is undeliverable. When the address is unassigned, the check digit is based on the unverified input Postcode1 (ZIP Code).</p>
Country	The country name. This transform does not attempt to make an assignment for addresses outside of the U.S. and its possessions, territories, and protectorates.
County_Code	The three-digit county code. Numbers start at 001 within each state.
Data_Source_ID	<p>The input source or list identifier.</p> <p>Use this field to identify the source of an input set or to identify the list that an input record belongs to in the case that multiple lists are present in the input.</p> <p>Statistics are generated for each unique value in this field when you map the field in conjunction with enabling the Gather Statistics Per Data Source option in the Reports and Statistics group.</p>
Delivery_Point	The two-digit DPBC code.
Family_Name1	The family name (for example, Smith).
Firm	The company name.
Given_Name1	The given name (for example, Robert).
Given_Name2	The second given name (for example, B.).

Input field name (USA Regulatory Address Cleanse)

Description

Lastline	The last line delivery information that can include all or some of the following fields: Locality1, Region1, Postcode1, or Postcode2.
Locality1	The city, town, or suburb.
Locality2	The Puerto Rican urbanization information.
LOT	The Line-of-Travel number.
LOT_Order	<p>The Line-of-Travel sortation:</p> <p><i>A</i>: Ascending</p> <p><i>D</i>: Descending</p> <p>LOT codes are required for non-automated, CART presorting in Standard Mail, Enhanced Carrier Route Subclass.</p>
Multiline1-12	A line from the input file which may contain data. The type of data in this line may vary from record to record.
Name	The name of the person associated with the address.
Postcode_Full	The complete postal code (ZIP10 with a hyphen; ZIP9 without a hyphen).
Postcode1	The five-digit primary ZIP Code. It does not include the four-digit ZIP4 Code.
Postcode2	The four-digit ZIP4 code. On a mail piece, this code follows the primary postal code with a hyphen placed between; for example, 54601-1234.
Postname	The honorary postname (indicating certification, academic degree, or affiliation such as CPA) or maturity postname (indicating heritage such as Jr.).
Prename	The prename (for example, Mr.).
Region1	The name of the state or province for this address.
SortCode_Route	The four-digit carrier route number.
Stage_Address_Flag	The USA Regulatory Address Cleanse information required from the stage file. For NCOALink stage testing only.
Stage_Lastline_Flag	
Stage_Name_Flag	
Stage_Record_Key	
Suggestion_Reply1-5	<p>The index number that corresponds to a specific lastline suggestion or an address line suggestion. These fields can also be used to input a street primary range or a street secondary range.</p> <p>If you do not want to use a suggestion list, make the value of this field 0, and the suggestion list will be ignored.</p>
Unit_Number	The secondary address information (for example, the unit description and/or secondary number).

8.5.12.15.3 Output fields for the USA Regulatory Address Cleanse transform

The following are Data Services output fields that can be used for the USA Regulatory Address Cleanse transform. See the fields listed in the transform's [Output](#) tab to view each field's properties.

Output field name (USA Regulatory Address Cleanse)	Description
Address_Line	<p>Complete, standardized primary and secondary address line. The style of suffixes, directional, and unit designators depends on how you define your options.</p> <div><p>i Note</p><p>If the output values don't fit within the length of the output field, then intelligent truncation occurs.</p></div>
Address_Line_Remainder1	<p>Extraneous data found on the address line that cannot be identified as address data or does not belong in the standardized address line. This may include invalid secondary address information.</p>


Output field name (USA Regulatory Address Cleanse)

Description

Address_Type (DELIVERY, DUAL)	<p>The record-type indicator for the assigned address. Applicable for DELIVERY and DUAL Generated Field Address Class.</p> <p>The first character indicates the type of record in the address directory to which the address matched:</p> <p><i>F</i>: Firm</p> <p><i>G</i>: General delivery</p> <p><i>H</i>: High-rise apartment or office building</p> <p><i>M</i>: Military</p> <p><i>P</i>: Post office box</p> <p><i>R</i>: Rural route or highway contract</p> <p><i>S</i>: Street (usually, one side of one city block)</p> <p><i><blank></i>: Unassigned</p> <p>The second character may be a <i>D</i> or blank. The D stands for default; it means that the transform detected, from the address directory, that a finer level of address assignment would be possible if further input information were available.</p> <p><i>FD</i>: Firm default. The transform did not assign a firm-level Postcode2, but could do so if given more or better firm information.</p> <p><i>GD</i>: General delivery default. Assigned when General Delivery is the only primary name listed for the Postcode1.</p> <p><i>HD</i>: High-rise default. The transform assigned the Postcode2 for the entire building. Assignment at the unit, floor, or wing level is possible. Often caused by a suite or apartment number out of range.</p> <p><i>RD</i>: Rural route or highway contract default. The transform assigned the Postcode2 for the entire route but could make a better assignment with the box number.</p> <p><i>SD</i>: Street default. Usually means that there is no Postcode2 for the building, so the transform had to assign the Postcode2 for the block.</p> <p><i>UD</i>: Unique default. Either the owner of the unique Postcode1 has not provided Postcode2 assignments, or the address could not be matched.</p> <p>When the transform cannot assign an address, it provides an address-type indication based on the way that the input data was parsed. This process is not foolproof. The transform may indicate that a street, rural route, highway contract, general delivery, or PO Box was parsed.</p>
AGeo_Countycode	Federal Information Processing Standard (FIPS) county code (for example, 063).
AGeo_Latitude	Latitude (degrees north of the equator) in the format 12.123456.
AGeo_Longitude	Longitude (degrees west of the Greenwich Meridian) in the format -12.123456.
AGeo_MCDCode	U.S. Census Bureau Minor Civil Division (MCD) data or if MCD data is unavailable, Census County Division (CCD) data (for example, 40775).

Output field name (USA Regulatory Address Cleanse)

Description

AGeo_Placecode	Federal Information Processing Standard (FIPS) place code (for example, 40775).
AGeo_Sectioncode	U.S. census tract code in the format 1234567890 (for example, 0003001059).
AGeo_Statecode	Federal Information Processing Standard (FIPS) state code (for example, 55).
Alias_Type (OFFICIAL)	<p>The alias-type indicator for the assigned address. Applicable for OFFICIAL Generated Field Address Class. It describes the input address, not the output address.</p> <p><i>A</i>: The input address matches an abbreviated street name.</p> <p><i>B</i>: The input address matches the high-rise alternate default base record.</p> <p><i>C</i>: The input street name is out of date; to get new street name, convert your record to the preferred alias.</p> <p><i>H</i>: The input address is an undesirable alternate, subject to conversion to a USPS preferred street address (high-rise alternate).</p> <p><i>O</i>: The input address is a street nickname or other alias.</p> <p><i>P</i>: The input address is a preferred alias.</p> <p><i><blank></i>: The input address is not an alias or is unassigned.</p>
ANKLink_Return_Code	<p>ANKLink return code (Attempted Not Known). Valid values are:</p> <p><i>77</i>: An ANKLink match was found. If NCOALink_Return_Code contains an A, 91, or 92, you may be able to obtain a new address from an NCOALink full service provider.</p> <p><i><blank></i>: No NCOALink lookup or no ANKLink match. This is always blank for full service providers.</p>
Audit_Dropped_Secondary	This field is used for audit testing. This field is also populated when an ANKLink match is made.
Audit_Prename	<p>These fields contain the name data used to make an NCOALink match. In some cases, the name in these fields is not the same as the input name (for example, if a nickname, alternate spelling, or initial is used instead).</p> <p>In the case of a firm match, these name fields contain a split version of the firm data.</p> <p>These fields are also populated when an ANKLink match is made.</p>
Audit_Given_Name1	
Audit_Given_Name2	
Audit_Family_Name1	
Audit_Postname	
Audit_Gender	This field is used for audit testing. This field is also populated when an ANKLink match is made.
Audit_General	<p>This field contains information for Stage I and Stage II tests, specifically query data, result data, and hint bytes, as the USPS requires. Use this field for audit purposes only. This field is required for audits.</p> <p>For more information about the content of this field, see the NCOALink User Technical Reference at http://ribbs.usps.gov/ .</p> <p>This field can also contain ANKLink return codes.</p>
Audit_Primary_Name	This is the primary name that is sent to NCOALink for matching. This field is required for audits.

Output field name (USA Regulatory Address Cleanse)	Description
Audit_Range	This is the range that is sent to NCOALink for matching. This field is required for audits.
Audit_Secondary_Range	This is the secondary range that is sent to NCOALink for matching. This field is required for audits.
Audit_Truncated_Given_Name1	This field is used for audit testing. This field is also populated when an ANKLink match is made.
Audit_Truncated_Given_Name2	This field contains the truncated middle name as stored in the NCOALink data. Use this field for audit purposes only. This field is required for audits.
Audit_Unit	This is the unit data that is sent to NCOALink for matching. This field is required for audits.
Carrier_Route_Sort_Zone	<p>The carrier-route sort zone indicates eligibility for Standard Mail Automation Enhanced Carrier Route.</p> <p><i>A</i>: Carrier route rates are available and merging is allowed.</p> <p><i>B</i>: Carrier route rates are available and merging is not allowed.</p> <p><i>C</i>: Carrier route rates are not available and merging is allowed.</p> <p><i>D</i>: Carrier route rates are not available and merging is not allowed.</p>
CASS_Assignment_Type	<p>Indicates the option used in making the assignment:</p> <p><i>O</i>: The non-CASS and DPV tie-break options are disabled or not used to make an assignment.</p> <p><i>1</i>: Inexact Postcode1 move assignment.</p> <p><i>2</i>: Input Postcode2 assignment.</p> <p><i>3</i>: DPV tie-breaking was used to make this assignment.</p> <p><i><blank></i>: The transform cannot assign an input address.</p>
CASS_Record_Type	<p>The record type necessary for posting on the CASS test. This field is populated for assigned records only. The valid record types include:</p> <p><i>F</i>: Firm</p> <p><i>G</i>: General delivery</p> <p><i>H</i>: High-rise</p> <p><i>P</i>: Post office box</p> <p><i>R</i>: Rural route or highway contract</p> <p><i>S</i>: Street</p>

Output field name (USA Regulatory Address Cleanse)

Description

CGeo_BSACode	<p>A Core-Based Statistical Area (CBSA) that consists of:</p> <ul style="list-style-type: none"> • A county with an incorporated place or a census-designated place that has a population of at least 10,000. • Adjacent counties with at least 25 percent of employed residents of the county who work in the CBSA's core or central county. <p>For example, 29100. CBSAs are either metropolitan (population of at least 50,000) or micropolitan (population between 10,000 and 50,000). With CBSAs, you can collect statistics for less urban areas of the country. CBSAs cover approximately 90 percent of the entire U.S. population.</p>
CGeo_Latitude	Latitude (degrees north of the equator) in the format 12.123456.
CGeo_Longitude	Longitude (degrees west of the Greenwich Meridian) in the format -12.123456.
CGeo_Metrocode	Metropolitan Statistical Area (MSA) number (for example, 3870). The value 0000 indicates that the address does not lie in any MSA (usually a rural area).
CGeo_Sectioncode	<p>U.S. census tract and block group code (for example, 0202001067). The first six digits are the tract number, and the first of the final four digits is the block group code within the tract.</p> <p>The Metropolitan Statistical Area (MSA) and block group codes are used for matching to demographic-coding databases. To uniquely specify a census block group within the entire country, combine the Sortcode_Postcode and CGeo_Sectioncode fields.</p>
Check_Digit	Check digit for the delivery-point bar code, or for a five-digit bar code if a full postal code (ZIP+4) could not be assigned.
Count	Specifies the suggestion count generated as the result of looking up the current record. A nonnegative value is output. If the current record does not end processing with a suggestion list needing resolution, then the value in this field is 0.
Country	The country name.
County_Code	Federal Information Processing Standard (FIPS) three-digit county code. Numbers are unique within states. You might use county information if you are preparing a presorted periodicals mailing.
County_Name	The fully-spelled county name.
Delivery_Point	The two-digit DPBC code.

Output field name (USA Regulatory Address Cleanse)

Description

Delivery_Type	<p>Type of postal facility:</p> <p><i>A</i>: Airport Mail Facility (AMF)</p> <p><i>B</i>: Branch Office</p> <p><i>C</i>: Community Post Office (CPO)</p> <p><i>D</i>: Area Distribution Center (ADC)</p> <p><i>E</i>: Sectional Center Facility (SCF)</p> <p><i>F</i>: Delivery Distribution</p> <p><i>G</i>: General Mail Facility (GMF)</p> <p><i>K</i>: Network Distribution Centers (NDC)</p> <p><i>M</i>: Money Order Unit</p> <p><i>N</i>: City/place name</p> <p><i>P</i>: Post Office (main)</p> <p><i>S</i>: Station</p> <p><i>U</i>: Urbanization (Puerto Rico only)</p>
District	District number for the U.S. House of Representatives.
DPV_CMRA	<p>The DPV Commercial Mail Receiving Agencies (CMRA) component that is generated for this record.</p> <p><i>L</i>: The address triggered DPV locking.</p> <p><i>N</i>: The address is not a CMRA.</p> <p><i>Y</i>: The address is a valid CMRA.</p> <p><i><blank></i>: A blank output value indicates that Enable_DPV_Validation is set to No, DPV processing is currently locked, or the transform cannot assign the input address.</p>

Output field name (USA Regulatory Address Cleanse)

Description

DPV_Footnote

DPV footnotes are required for end-user CASS certification. The footnotes contain the following information:

AA: The input address matches to the ZIP+4 file.

A1: The input address does not match to the ZIP+4 file.

BB: All input address field values match to DPV.

CC: The input address primary number matches to DPV, but the secondary number does not match (the secondary is present but invalid).

F1: The input address matches to a military address.

G1: The input address matches a general delivery address.

M1: The input address primary number is missing.

M3: The input address primary number is invalid.

N1: The input address primary number matches to DPV but the address is missing the secondary number.

NL: An NCOALink move address cannot be DPV confirmed. The NCOALink directory data does not exactly match the DPV directory data. This may happen because the NCOALink directories are updated more frequently than the DPV directories.

i Note

The NL footnote is applicable only for the Move Updated generated field class.

P1: The input address is missing the rural route or highway contract box number.

P3: The input address is an invalid post office, rural route, or highway contract number.

R1: The input address matches to CMRA, but the secondary number is not present.

RR: The input address matches to CMRA.

UI: The input address matches a unique address.

i Note

The transform always posts the DPV footers in the same order and this field is not always 12 characters in length.

Output field name (USA Regulatory Address Cleanse)

Description

DPV_NoStats	<p>No Stat indicator. No Stat means that the address is a vacant property, it receives mail as a part of a drop, or it does not have an established delivery yet.</p> <p><i>Y</i>: The address is flagged as No Stat in DPV data.</p> <p><i>N</i>: The address is not No Stat.</p> <p><i><blank></i>: The address was not looked up.</p> <div> <p>i Note</p> <p>The US Addressing report contains DPV NoStats counts in the DPV Summary section.</p> </div>
DPV_Status	<p>The DPV status component that is generated for this record.</p> <p><i>D</i>: The primary range is a confirmed delivery point, but the secondary range is not available on input.</p> <p><i>L</i>: The address triggered DPV locking.</p> <p><i>N</i>: The address is not a valid delivery point.</p> <p><i>S</i>: The primary range is a valid delivery point, but the parsed secondary range is not valid in the DPV directory.</p> <p><i>Y</i>: The address is a confirmed delivery point. The primary range and secondary range (if present) are valid.</p> <p><i><blank></i>: A blank output value indicates that Enable_DPV_Validation is set to No, DPV processing is currently locked, or the transform cannot assign the input address.</p>
DPV_Vacant	<p>Vacant address indicator.</p> <p><i>Y</i>: The address is vacant.</p> <p><i>N</i>: The address is not vacant.</p> <p><i><blank></i>: The address was not looked up.</p> <div> <p>i Note</p> <p>The US Addressing report contains DPV Vacant counts in the DPV Summary section.</p> </div>
DSF2_Business_Indicator	<p>Residential/business indicator. You may use this information to lower your parcel-shipping costs. (Some parcel delivery services charge more for delivery to residential addresses.)</p> <p><i>Y</i>: Business address.</p> <p><i>N</i>: Not a business address.</p> <p><i><blank></i>: The address was not looked up.</p>

Output field name (USA Regulatory Address Cleanse)

Description

DSF2_Delivery_Type	<p>Delivery type.</p> <p><i>1</i>: Curb-side delivery.</p> <p><i>2</i>: NDCBU (Neighborhood Delivery Centralized Box Unit) delivery.</p> <p><i>3</i>: Central delivery.</p> <p><i>4</i>: Door-slot delivery.</p> <p><blank>: The address was not looked up.</p>
DSF2_Drop_Count	<p>Drop count.</p> <p>If DSF2_Drop_Indicator contains Y or DPV_CMRA contains Y, then this field contains a value from 000 to 999, indicating the number of businesses or families served by this delivery point.</p>
DSF2_Drop_Indicator	<p>Drop indicator.</p> <p><i>Y</i>: The delivery point serves multiple businesses or families. For example, delivery point may be a CMRA (Commercial Mail Receiving Agency).</p> <p><i>N</i>: The delivery address is not a CMRA.</p> <p><blank>: The address was not looked up.</p>
DSF2_Educational_Ind	<p>Educational indicator.</p> <p><i>Y</i>: The address is an educational institution.</p> <p><i>N</i>: The address is not an educational institution.</p> <p><blank>: The address was not looked up.</p>
DSF2_LACS_Conversion_Indicator	<p>LACS (Locatable Address Conversion System) indicator.</p> <p><i>Y</i>: The address is LACS convertible.</p> <p><i>N</i>: The address is not LACS convertible.</p> <p><blank>: The address was not looked up.</p>
DSF2_Record_Type	<p>Record type.</p> <p><i>B</i>: Business address.</p> <p><i>R</i>: Residential address.</p> <p><i>U</i>: Unknown. AP.DSF_Deltype is blank.</p> <p><blank>: No information available.</p>
DSF2_Seasonal_Indicator	<p>Seasonal address indicator.</p> <p><i>Y</i>: The address is seasonally occupied.</p> <p><i>N</i>: The address is not seasonal.</p> <p><blank>: The address was not looked up.</p>

Output field name (USA Regulatory Address Cleanse)

Description

DSF2_Throwback_Indicator	<p>Throwback indicator.</p> <p><i>Y</i>: Customer with street address wants delivery at PO Box instead.</p> <p><i>N</i>: No throwback necessary.</p> <p><i><blank></i>: The address was not looked up.</p>
Error	<p>Specifies the error status generated as the result of looking up the current record and performing a suggestion processing. Possible output values are 0–5.</p> <p><i>0</i>: There are no suggestion selection errors.</p> <p><i>1</i>: The necessary selection information is blank. For example, a lastline suggestion list is generated, but there is no lastline selection input field data to make a choice.</p> <p><i>2</i>: The suggestion selection is invalid. For example, 8 was selected but there are only five suggestions.</p> <p><i>3</i>: The suggestion entry in the input field is invalid.</p> <p><i>4</i>: The suggestion range in the input field is invalid.</p> <p><i>5</i>: The suggestion secondary range in the input field is invalid.</p>
EWS_Match	<p>Returns the results of the EWS (Early Warning System) match.</p> <p><i>T</i>: True, the address is located in the EWS directory and is an EWS match.</p> <p><i>F</i>: False, the address is not located in the EWS directory.</p> <p><i><blank></i>: EWS is not enabled.</p>
Extra1-10	<p>Any non-address data found above or below the address data in the address block. Available only if the input data is presented through multiline fields.</p>
Extraneous_Secondary_Address_Data	<p>Consists of the data from Extraneous_Secondary_Unit_Number and Extraneous_Secondary_Non_Postal respectively. Any additional # data is placed in the remainder or extra components. This may include invalid secondary address data.</p>
Extraneous_Secondary_Non_Postal	<p>Extraneous data retained in this field is the best guess at Private Mail Box data, based on the position in the address line and other information contained in the address, such as a pound unit designator (#). This may include invalid secondary address data.</p>
Extraneous_Secondary_Unit_Number	<p>Extraneous data retained in this field is the best guess at secondary range data, based on the position in the address line and other information contained in the address. This may include invalid secondary address data.</p>
Fault_Code	<p>The fault code. This is blank if the address is assigned.</p> <p>For more information, see Fault codes (USA Regulatory Address Cleanse) [page 815].</p>
Fault_Or_Status_Code	<p>The fault code if the address is unassigned; the status code if the address is assigned.</p>
Finance_Area_Postcode	<p>The Finance Area Postcode is the lowest Postcode1 within a Finance Number. (Finance Numbers are currently used to link data to a single post office or postmaster.)</p>

Output field name (USA Regulatory Address Cleanse)

Description

Firm	<p>Firm name. Do not use this field if the input was multiline, because if there is no firm name in the postal directory, the transform cannot reliably identify firm names from multilines.</p> <p>If you retrieve the corrected component, the firm name is taken from the postal directory if found; otherwise, it's taken from the input record. Be aware that the postal directory might contain some unusual or shortened spellings that you may or may not find suitable for printing on mail pieces. If you prefer to retain your own firm data, retrieve the original component.</p>
Foreign_Code	<p>Specifies whether the address is foreign or domestic.</p> <p><i>F</i>: Foreign addresses</p> <p><i><blank></i>: Domestic U.S.</p>
Full_Address	<p>The complete address line, including secondary address and dual address (street and postal) line data.</p> <div><p>i Note</p><p>This field may contain invalid secondary address information when you set the <i>Include Unused Address Line Data</i> option in the Standardization Options group to <i>Yes</i>.</p></div> <div><p>i Note</p><p>If the output values don't fit within the length of the output field, then intelligent truncation occurs.</p></div>

Output field name (USA Regulatory Address Cleanse)

Description

Geo_Matchcode	Match code indicating the precision of the latitude and longitude assignment.																																																						
	<table><tr><th>Value</th><th>Description</th><th>Centroid</th><th>Address</th><th>Best</th><th>All</th></tr><tr><td>0</td><td>Address-level match</td><td></td><td>X</td><td>X</td><td>X</td></tr><tr><td>1</td><td>9-digit centroid-level match</td><td>X</td><td></td><td>X</td><td></td></tr><tr><td>4</td><td>7-digit centroid-level match</td><td>X</td><td></td><td>X</td><td></td></tr><tr><td>5</td><td>5-digit centroid-level match</td><td>X</td><td></td><td>X</td><td></td></tr><tr><td>7</td><td>No centroid-level match</td><td>X</td><td></td><td></td><td></td></tr><tr><td>8</td><td>No address-level match</td><td></td><td>X</td><td></td><td>X</td></tr><tr><td>9</td><td>Both address- and centroid-level tried; no match in either</td><td></td><td></td><td>X</td><td></td></tr><tr><td><blank></td><td>No lastline match</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>	Value	Description	Centroid	Address	Best	All	0	Address-level match		X	X	X	1	9-digit centroid-level match	X		X		4	7-digit centroid-level match	X		X		5	5-digit centroid-level match	X		X		7	No centroid-level match	X				8	No address-level match		X		X	9	Both address- and centroid-level tried; no match in either			X		<blank>	No lastline match	X	X	X	
Value	Description	Centroid	Address	Best	All																																																		
0	Address-level match		X	X	X																																																		
1	9-digit centroid-level match	X		X																																																			
4	7-digit centroid-level match	X		X																																																			
5	5-digit centroid-level match	X		X																																																			
7	No centroid-level match	X																																																					
8	No address-level match		X		X																																																		
9	Both address- and centroid-level tried; no match in either			X																																																			
<blank>	No lastline match	X	X	X																																																			
Intermediate_Codes	Intermediate codes provide information that the USPS requires when you perform NCOALink certification or audit testing.																																																						
LACSCode	<p>LACS (Locatable Address Conversion System) indicator.</p> <p><i>T</i>: The address needs 9-1-1 conversion (from box to street address) and should be submitted to a LACS vendor.</p> <p><i>F</i>: The address does not need conversion.</p> <p><blank>: The address was not assigned.</p>																																																						
LACSLink_Indicator	<p>Returns the conversion status of addresses processed by LACSLink.</p> <p><i>Y</i>: The address was converted by LACSLink (the LACSLink_Return_Code value is A).</p> <p><i>N</i>: The address was looked up with LACSLink but not converted.</p> <p><i>F</i>: The address was a false-positive.</p> <p><i>S</i>: A LACSLink conversion was made, but it was necessary to drop the secondary information.</p> <p><blank>: No LACSLink lookup attempted.</p>																																																						

Output field name (USA Regulatory Address Cleanse)

Description

LACSLink_Query	<p>Returns the pre-conversion address, populated only when LACSLink is turned on and a LACSLink lookup was attempted. This address is in the standard Pub. 28 format. However, when an address has both a unit designator and secondary unit, the unit designator is replaced by the pound character (#).</p> <p><blank>: No LACSLink lookup attempted.</p>
LACSLink_Return_Code	<p>Returns the match status for LACSLink processing.</p> <p><i>A</i>: LACSLink record match. A converted address is provided in the address data fields.</p> <p><i>00</i>: No match and no converted address.</p> <p><i>09</i>: LACSLink matched an input address to an old address, which is a "high-rise default" address; no new address is provided.</p> <p><i>14</i>: Found a LACSLink record, but couldn't convert the data to a deliverable address.</p> <p><i>92</i>: LACSLink record matched after dropping the secondary number from input address.</p> <p><blank>: No LACSLink lookup attempted.</p>
Lastline	<p>Locality, region, and postal code together on one line.</p> <div> <p>i Note</p> <p>If the output values don't fit within the length of the output field, then intelligent truncation occurs.</p> </div>
Locality1	<p><i>Canada and USA engines</i>: Locality preferred by the postal authority.</p> <p><i>Other engines</i>: City, town, locality, or suburb.</p> <div> <p>i Note</p> <p>If the output values don't fit within the length of the output field, then intelligent truncation occurs.</p> </div>
Locality1_Alternate	<p>Preserves the input Locality1 if it is not recognized by the postal authority as a valid Locality1 name for the input address line, and the original Locality1 was changed because of address assignment rules.</p> <p>Also outputs the default Locality 1 for the assigned finance area when there is no input Locality1, or the input Locality1 is not valid in the assigned finance area.</p>
Locality1_LLIDX	<p>Yields a city name (locality1 name) that is more geographically precise than Locality1_Official.</p> <p>LLIDX (last-line index) is a USPS number that ties a ZIP+4 record to a particular city, state, and ZIP.</p> <div> <p>i Note</p> <p>If the output values don't fit within the length of the output field, then intelligent truncation occurs.</p> </div>

Output field name (USA Regulatory Address Cleanse)

Output field name (USA Regulatory Address Cleanse)	Description
Locality1_Name	<p>The city, town, or suburb.</p> <p>Locality names that are marked as invalid for mailing by the USPS are always preserved, never converted, regardless of the values set for the Preserve Place Name and Assign With Input Locality options.</p>
Locality1_Official	<p>The standardized locality name. When the input city name is tagged by the USPS as invalid for mailing, this field always yields a converted city name, no matter how the Preserve Place Name option is set.</p>
Locality1_Official_ABBR	<p>The official USPS abbreviation of the city name, if one is available. This field is blank if the full city name is less than 13 characters or if the full name is longer, but the USPS has not provided an official abbreviation.</p>
Locality2	<p>USA engine: Urbanization (Puerto Rican addresses only).</p> <p>Other engines: Additional city, town, locality, or suburb information.</p>
Locality2_Official	<p>Urbanization name; produced only when the address is in Puerto Rico.</p>
LOT	<p>Line-of-travel number.</p>
LOT_Order	<p>Line-of-travel sortation.</p> <p>A: Ascending.</p> <p>D: Descending.</p>
Matched_Addressline_Indicator	<p>Match level indicator.</p> <p>T: The address line is matched to a ZIP+4 record.</p> <p>F: The address line is not matched to a ZIP+4 record.</p>
Matched_Lastline_Indicator	<p>Match level indicator.</p> <p>T: The last line is matched to a City/ZCF record.</p> <p>F: The last line is not matched to a City/ZCF record.</p>
Move_Effective_Date	<p>The date that the move is effective as indicated on the change of address card sent to the USPS in the format yyyyymm. The yyyyymm format is returned from the NCOALink directories and is required by the USPS for audit purposes.</p> <p>To use it in a function or post it to an output file, you'll probably have to convert the format to mm/dd/yyyy first.</p> <p>This field is also populated when an ANKLink match is made.</p>
Move_Type	<p>Type of move record.</p> <p>B: Business (matched by company name).</p> <p>F: Family (matched by last name).</p> <p>I: Individual (matched by first and last name).</p> <p>This field is also populated when an ANKLink match is made.</p>

Output field name (USA Regulatory Address Cleanse)

Description

Multiline1-12	<p>A line that may contain any data. The type of data in this line may vary from record to record.</p> <div> <p>i Note</p> <p>These fields may contain invalid secondary address information when you set the Include Unused Address Line Data option in the Standardization Options group to Yes.</p> </div> <div> <p>i Note</p> <p>For address data, if the output values don't fit within the length of the output field, then intelligent truncation occurs.</p> </div>
Name	The name of a person associated with the address.
NCOALink_Hint_Byte	This field is used for audit testing.
NCOALink_Return_Code	<p>This field shows NCOALink return codes. To populate this field, set the List Processing Mode to one of the three available options: Change of Address, Statistics Only, or Return Codes Only.</p> <p>A brief description of the return codes appears on the NCOALink Processing Summary report. To print more detailed return code descriptions on the report, enable the Generate Return Code Descriptions option in the NCOALink Report Options group.</p> <p>This field is also populated when an ANKLink match is made.</p>
Stage_Test_Record	<p>This field is for stage testing only. It applies to NCOALink, CASS, DSF2 Augment, DSF2 Sequence, and DSF2 Invoice self-certifications.</p> <p>The USA Regulatory Address Cleanse transform populates the values of this field automatically to match the format required for stage testing.</p>
Non_CASS_Firm	The firm match that is made by using the input ZIP+4 for missing or invalid firm information.
Non_CASS_Secondary_Address	<p>The secondary address match that is made using the input ZIP+4 for missing or invalid secondary address information.</p> <div> <p>i Note</p> <p>If the output values don't fit within the length of the output field, then intelligent truncation occurs.</p> </div>
Non_CASS_Unit	The unit designator match that is made using the input ZIP+4 for missing or invalid unit designator information.
Non_CASS_Unit_Number	The unit designator match that is made using the input ZIP+4 for missing or invalid unit designator information.
Non_Postal_Secondary_Address	The complete non-postal secondary address (for example, "PMB 10" or "# 10"). Non-postal means that the mail is delivered through a private mailbox company rather than the USPS.
Non_Postal_Unit	Non-postal unit designator (PMB or #). Non-postal means that the mail is delivered through a private mailbox company rather than the USPS.

Output field name (USA Regulatory Address Cleanse)

Output field name (USA Regulatory Address Cleanse)	Description
Non_Postal_Unit_Number	Non-postal secondary range (PMB number only, does not include designator). Non-postal means that the mail is delivered through a private mailbox company rather than the USPS.
Parsed_Firm	<p>If the change of address is made based on a firm (company) name, the firm name is posted in this field.</p> <p>This field is also populated when an ANKLink match is made.</p>
Postal_Box_Number	Post office box number.
Postcode_Full	The complete ZIP10 with a hyphen.
Postcode_Full_No_Hyphen	The complete ZIP9 without a hyphen.
Postcode_Type	<p>The type of ZIP Code that is assigned.</p> <p><i>M</i>: Military.</p> <p><i>U</i>: Unique (specific to a university, large firm, or other institution).</p> <p><blank>: Ordinary ZIP Code or the ZIP Code was not assigned.</p>
Postcode1	The five-digit ZIP Code. Does not include the four-digit ZIP4.
Postcode1_Change_Ind	<p>Indicates whether the address is affected by postal code realignment.</p> <p><i>T</i>: True, the transform corrected the postal code (and the locality, if applicable).</p> <p><i>F</i>: False.</p> <p><blank>: The address was not corrected.</p>
Postcode2	Four-digit ZIP4 Code. On a mail piece, this code follows the primary postal code, with a hyphen placed between, for example, 54601-1234.
Pre_Suitelink_Delivery_Point	The numeric two-digit code for the delivery point bar code that was generated before SuiteLink processing.
Pre_Suitelink_Postcode1	<p>The ZIP Code that was assigned by the transform before SuiteLink processing.</p> <p><i>5-digit ZIP Code</i>: SuiteLink Retcode value is A.</p> <p><blank>: No ZIP Code assigned.</p>
Pre_Suitelink_Postcode2	The ZIP+4 that was assigned by the transform before SuiteLink processing. The ZIP+4 is either for a high-rise default or street default record.
Pre_Suitelink_Unit_Description	The unit designator that existed before SuiteLink processing. If this field is blank, the transform did not assign any secondary information.
Pre_Suitelink_Unit_Number	The secondary range information that existed before SuiteLink processing. If this field is blank, the transform did not assign any secondary information.
Primary_Address	<p>Primary address line, such as the street address or post office box. Does not include secondary address information such as apartment.</p> <p>If the <i>Use USPS Primary Name Abbreviation</i> option is enabled, the software uses the USPS Primary Name abbreviation first. If the values don't fit within the length of the output fields, then intelligent truncation occurs.</p>

Output field name (USA Regulatory Address Cleanse)

Description

Primary_Name1	<p>Street name description.</p> <div> <p>i Note</p> <p>If the output values don't fit within the length of the output field, then intelligent truncation occurs.</p> </div>
Primary_Number	The premise number.
Primary_Postfix1	An abbreviated directional (such as N, S, NW, or SE) that follows a street name.
Primary_Postfix1_Long	A fully-spelled directional (such as North or South) that follows the street name.
Primary_Prefix1	An abbreviated directional (such as N, S, NW, or SE) that precedes a street name.
Primary_Prefix1_Long	A fully-spelled directional (such as North or South) that precedes the street name.
Primary_Secondary_Address	<p>The primary address and secondary address on one line. Does not include remainder data. This line is always output as if the Include Unused Address Line Data option is set to No, which means that the output does not include invalid secondary address line information.</p> <div> <p>i Note</p> <p>If the output values don't fit within the length of the output field, then intelligent truncation occurs.</p> </div>
Primary_Type1	Abbreviated street type (for example, St, Ave, or Pl).
Primary_Type1_Long	Fully-spelled street type (for example, Street or Avenue).
QSS_Default	<p>Specifies whether the record qualified as a default match instead of qualifying as a match at a higher level of assignment. Output values are:</p> <p>T: True</p> <p>F: False</p>
RDI_Indicator	<p>The residential delivery indicator (RDI) shows whether the address is residential or nonresidential.</p> <p>Y: Residential address</p> <p>N: Nonresidential address</p>
Region1	State, province, territory, or region.
Rural_Route_Box_Number	The rural route box number.
Rural_Route_Number	The rural route number.
Secondary_Address	The building name, floor, and room number in one field.
Sortcode_Postcode	The federal code for state and county (FIPSCODE). Combines the two-digit state code with the three-digit county code. U.S. territories, possessions, or protectorates such as Puerto Rico, the U.S. Virgin Islands, or the Pacific Islands do not have FIPS state digits.
Sortcode_Route	The four-digit carrier route.

Output field name (USA Regulatory Address Cleanse)	Description
Status	<p>Specifies the suggestion status generated as the result of looking up the current record and performing suggestion processing.</p> <p><i>A</i>: Suggestion processing ended with an address suggestion list needing resolution.</p> <p><i>L</i>: Suggestion processing ended with a lastline suggestion list needing resolution.</p> <p><i>N</i>: No suggestion lists were generated and no suggestion processing was performed.</p> <p><i>R</i>: The primary range is invalid for the selected address suggestion.</p> <p><i>S</i>: The secondary range is invalid for the selected address suggestion.</p> <p><i>U</i>: The secondary address is invalid for the selected address suggestion.</p>
Status_Code	<p>The status code. This field is blank if the address is unassigned.</p> <p>For more information, see Status codes (USA Regulatory Address Cleanse) [page 816].</p>
Suggestion_List	Contains all of the Suggestion List Component field values that you chose in the Suggestion List group of the USA Regulatory Address Cleanse transform.
SuiteLink_Retcode	<p><i>A</i>: SuiteLink match—Secondary information exists and was assigned to this record as a result of SuiteLink processing.</p> <p><i>00</i>: No SuiteLink match—Lookup was attempted but no matching record was found.</p> <p><i><blank></i>: A SuiteLink lookup was not attempted because one of the following is true:</p> <ul style="list-style-type: none"> • The address is not a high-rise default according to CASS. • The address does not contain a firm.
Undeliverable_Indicator	<p>Indicates whether the record is a deliverable address.</p> <p><i>T</i>: The address is tagged by the USPS as unsuitable for mail delivery (for example, a cemetery).</p> <p><i>F</i>: The address either was not matched to a ZIP+4 record or was matched to a record that indicates that the address is suitable for mail delivery.</p>
Unit_Description	Unit description (for example, #, Apartment, or Flat).
Unit_Description_Directory	Unit designator from ZIP+4 directory, or blank if none was found.
Unit_Number	Unit number (for example, 100 in "APT 100").

Related Information

[Fault codes \(USA Regulatory Address Cleanse\) \[page 815\]](#)

[Status codes \(USA Regulatory Address Cleanse\) \[page 816\]](#)

8.5.13 Address Cleanse reference

This section describes reference information for use with the Address Cleanse transforms (Global Address Cleanse and USA Regulatory Address Cleanse). For the USA Regulatory Address Cleanse transform, it lists the available certifications and the steps to achieve those certifications.

This section also explains how to use the Show A and Show L utilities (for United States addresses only) that you can use to query the postal directories used by either of the Address Cleanse transforms.

For the Global Address Cleanse transform:

- Information codes
- Status codes
- Quality codes

For the USA Regulatory Address Cleanse transform:

- Status codes
- Fault codes

Parent topic: [Data Quality transforms \[page 437\]](#)

Related Information

[Blueprints and other content objects for download \[page 438\]](#)

[About Data Quality fields \[page 445\]](#)

[About data quality statistics \[page 448\]](#)

[Associate \[page 454\]](#)

[Country ID \[page 472\]](#)

[Data Cleanse \[page 474\]](#)

[DSF2® Walk Sequencer \[page 514\]](#)

[Geocoder \[page 524\]](#)

[Global Address Cleanse \[page 552\]](#)

[Global Suggestion List \[page 640\]](#)

[Match \[page 651\]](#)

[USA Regulatory Address Cleanse \[page 710\]](#)

[Data Cleanse reference \[page 823\]](#)

8.5.13.1 Beyond the basic address cleansing

The USA Regulatory Address Cleanse transform offers many additional address cleanse features for U.S. addresses. These features extend address cleansing beyond the basic parsing and standardizing. To read about the USA Regulatory Address Cleanse transform and its options, see the *Reference Guide*.

8.5.13.1.1 USPS certifications

The USA Regulatory Address Cleanse transform is CASS-certified. Therefore, when you process jobs with the USA Regulatory Address Cleanse transform (and it is set up correctly) you reap the benefits of that certification.

If you integrate Data Services into your own software and you want to obtain CASS certification, you must follow the steps for CASS self-certification using your own software.

You can also obtain licenses for DSF2 (Augment, Invoice, Sequence) and for NCOALink by using USA Regulatory Address Cleanse and DSF2 Walk Sequencer blueprints that are specifically set up for that purpose.

i Note

In this section we direct you to the USPS website and include names of documents and procedures. The USPS may change the address, procedure, or names of documents (and information required) without our prior knowledge. Therefore some of the information may become outdated.

Related Information

[CASS self-certification \[page 769\]](#)

[DSF2 Certification \[page 777\]](#)

8.5.13.1.1.1 Completing USPS certifications

The instructions below apply to USPS CASS self-certification, DSF2 license, and NCOALink license certification.

During certification you must process files from the USPS to prove that your software is compliant with the requirements of your license agreement.

The CASS, DSF2, and NCOALink certifications have two stages. Stage I is an optional test which includes answers that allow you to troubleshoot and prepare for the Stage II test. The Stage II test does not contain answers and is sent to the USPS for evaluation of the accuracy of your software configuration.

1. Complete the applicable USPS application (CASS, DSF2, NCOALink) and other required forms and return the information to the USPS.
After you satisfy the initial application and other requirements, the USPS gives you an authorization code to purchase the CASS, DSF2, or NCOALink option.
2. Purchase the option from the USPS. Then submit the following information to SAP:
 - your USPS authorization code (see step 1)
 - your NCOALink provider level (full service provider, limited service provider, or end user) (applicable for NCOALink only)
 - your decision whether or not you want to purchase the ANKLink option (for NCOALink limited service provider or end user only)

After you request and install the feature from SAP, you are ready to request the applicable certification test from the USPS. The software provides blueprints to help you set up and run the certification tests. Import

them from `<DS_COMMON_DIR>\DataQuality\Certifications`, where `<DS_COMMON_DIR>` is the software's common configuration directory.

3. Submit the Software Product Information form to the USPS and request a certification test. The USPS sends you test files to use with the blueprint.
4. After you successfully complete the certification tests, the USPS sends you the applicable license agreement. At this point, you also purchase the applicable product from SAP.

Related Information

[Setting up the NCOALink blueprints \[page 775\]](#)

[Setting up the DSF2 certification blueprints \[page 779\]](#)

8.5.13.1.1.2 Introduction to static directories

Users who are self-certifying for CASS must use static directories. Those obtaining DSF2 licenses also need to use static directories. Static directories do not change every month with the regular directory updates. Instead, they can be used for certification for the duration of the CASS cycle. Using static directories ensures consistent results between Stage I and Stage II tests, and allows you to use the same directory information if you are required to re-test. You obtain static directories from SAP.

i Note

If you do not use static directories when required, the software issues a warning.

8.5.13.1.1.2.1 Static directories

The following directories are available in static format:

- `zip4us.dir`
- `zip4us.shs`
- `zip4us.rev`
- `revzip4.dir`
- `city10.dir`
- `zcf10.dir`
- `dpv*.dir`
- `elot.dir`
- `ew*.dir`
- SuiteLink directories
- LACSLink directories

8.5.13.1.1.2.2 Obtaining static directories

To request static directories, contact SAP Business User Support. Contact information (toll-free number and email address) is available at <https://support.sap.com/> .

1. Click [SAP Support Portal](#).
2. Click the [Help and Support](#) tab.
3. Click [Contact SAP](#).
4. Complete the form and then click [Submit](#).

8.5.13.1.1.2.3 Static directories location

It is important that you store your static directories separately from the production directories. If you store them in the same folder, the static directories will overwrite your production directories.

We suggest that you create a folder named “static” to store your static directories. For example, save your static directories under `<DS_COMMON_DIR>\DataQuality\reference\static`, where `<DS_COMMON_DIR>` is the software's common configuration directory.

8.5.13.1.1.2.4 Static directories safeguards

To prevent running a production job using static directories, the software issues a verification warning or error under the following circumstances:

- When the job has both static and non-static directories indicated.
- When the release version of the `zip4us.dir` does not match the current CASS cycle in the software.
- When the data versions in the static directories aren't all the same. For example, for CASS Cycle M the data versions in the static directories are M01.
- When the job is set for self-certification but is not set up to use the static directories.
- When the job is not set for self-certification but is set up to use the static directories.

8.5.13.1.1.3 Importing certification blueprints

The software includes blueprints to help you with certification testing. Additionally, the blueprints can be used to process a test file provided by the USPS during an audit. You need to first import the blueprints to the repository before you can use them in Data Services.

To import the certification blueprints, follow these steps:

1. Open Data Services Designer.
2. Right-click in the Object Library area and select  [Repository](#)  [Import from file](#) .
3. Go to `<DS_COMMON_DIR>\DataQuality\certifications`, where `<DS_COMMON_DIR>` is the software's common configuration directory.

4. Select the applicable blueprint and click [Open](#).

i Note

A message appears asking for a pass phrase. The blueprints are not pass phrase protected, just click [Import](#) to advance to the next screen.

5. Click [OK](#) at the message warning that you are about to import the blueprints.

Importing the blueprint files into the repository adds new projects, jobs, data flows, and flat file formats. The naming convention of the objects includes the certification type to indicate the associated certification test.

Related Information

[CASS self-certification blueprint \[page 770\]](#)

[NCOALink blueprints \[page 775\]](#)

[DSF2 Certification blueprints \[page 778\]](#)

8.5.13.1.1.4 CASS self-certification

If you integrate Data Services into your own software, and you want to CASS-certify your software, you must obtain CASS certification on your own (self certification). You need to show the USPS that your software meets the CASS standards for accuracy of postal coding and address correction. You further need to show that your software can produce a facsimile of the USPS Form 3553 . You need a USPS Form 3553 to qualify mailings for postage discounts.


Obtaining CASS certification on your own software is completely optional. However there are many benefits when your software is CASS certified.

Visit the USPS RIBBS website at <http://ribbs.usps.gov/index.cfm?page=cassmass>  for more information about CASS certification.

Related Information

[Completing USPS certifications \[page 766\]](#)

8.5.13.1.1.4.1 CASS self-certification process overview

1. Familiarize yourself with the CASS certification documentation and procedures located at <http://ribbs.usps.gov/index.cfm?page=cassmass>  .

2. (Optional.) Download the CASS Stage I test from the RIBBS website.
This is an optional step. You do not submit the Stage I test results to the USPS. Taking the Stage I test helps you analyze and correct any inconsistencies with the USPS-expected results before taking the Stage II test.
3. Import and make modifications to the CASS self-certification blueprint (`us_cass_self_certification.atl`). The blueprint is located in `<DS_COMMON_DIR>\DataQuality\Certifications`, where `<DS_COMMON_DIR>` is the software's common configuration location.
Edit the blueprint so it contains your static directories location, Stage I file location, your company name, and other settings that are required for CASS processing.
4. When you are satisfied that your Stage I test results compare favorably with the USPS-expected results, request the Stage II test from the USPS using the Stage II order form located on the RIBBS website.
The USPS will place the Stage II test in your user area on the RIBBS website for you to download.
5. Download and unzip the Stage II test file to an output area.
6. After you run the Stage II file with the CASS self-certification blueprint, check that the totals on the USPS Form 3553 and the actual totals from the processed file match.
7. Compress the processed Stage II answer file (using WinZip for example) and name it using the same name as the downloaded Stage II file (step 5).
8. Upload the processed Stage II answer file to your user area on the RIBBS website.

The USPS takes about two weeks to grade your test.

8.5.13.1.1.4.2 CASS self-certification blueprint

SAP provides a CASS self-certification blueprint. The blueprint contains the corresponding project, job, data flow, and input/output formats. Additionally, the blueprint can be used to process a test file provided by the USPS during an audit.

Import the `us_cass_self_certification.atl` blueprint from `<DS_COMMON_DIR>\DataQuality\Certifications` where `<DS_COMMON_DIR>` is the software's common configuration location. The following table contains the file names for the CASS self-certification blueprint:

Object	Name
ATL file	<code>us_cass_self_certification.atl</code>
Project	<code>DataQualityCertificationCASS</code>
Job	<code>Job_DqBatchUSAReg_CASSSelfCert</code>
data flow	<code>DF_DqBatchUSAReg_CASSSelfCert</code>
Input file format	<code>DqUsaCASSSelfCert_In</code>
Output file format	<code>DqUsaCASSSelfCert_Out</code>

8.5.13.1.1.4.3 USPS Form 3553 required options for self certification

The following options in the CASS Report Options group are required for CASS self certification. This information is included in the USPS Form 3553.

Option	Description
<i>Company Name Certified</i>	Specify the name of the company that owns the CASS-certified software.
<i>List Name</i>	Specify the name of the mailing list.
<i>List Owner</i>	Specify the name of the list owner.
	i Note Keep the CASS self-certification blueprints setting of "USPS".
<i>Mailer Address(1-4)</i>	Specify the name and address of the person or organization for whom you are preparing the mailing (up to 29 characters per line).
<i>Software Version</i>	Specify the software name and version number that you are using to receive CASS self certification.

8.5.13.1.1.4.4 Points to remember about CASS

Remember these important points about CASS certification:

- As an end user (you use Data Services to process your lists), you are not required to obtain CASS self certification because Data Services is already CASS certified.
- CASS certification is given to software programs. You obtain CASS self certification if you have incorporated Data Services into your software program.
- The CASS reports pertain to address lists.
- CASS certification proves that the software can assign and standardize addresses correctly.

8.5.13.1.1.5 NCOALink certification

NCOALink is a feature that must be purchased before testing.

The NCOALink certification consists of the following steps:

1. Application and Self-Certification Statement Approval
2. Software acquisition
3. Testing and certification
4. Execution of License Agreement

This entire procedure is documented in the USPS Certification Procedures documents posted on the RIBBS website at http://ribbs.usps.gov/ncoalink/documents/tech_guides . Select either NCOALink End User

Documents, NCOALink Limited Service Provider Documents, or NCOALink Full Service Provider Documents as applicable.

You must complete the appropriate service provider certification procedure for NCOALink in order to purchase the NCOALink product from the USPS.

8.5.13.1.1.5.1 About NCOALink directories

After you have completed the certification requirements and purchased the NCOALink product from the USPS, the USPS makes the latest NCOALink directories available monthly (if you're an end user licensee) or weekly (if you're a limited or full service provider licensee). Before you can use the NCOALink directories, you must download them (using the USPS downloading tool, EPF Downloader Manager, or another download interface) and then extract and uncompress them (using the SAP NCOALink Utility).

The SAP NCOALink Utility is a 64-bit application that is run from the command line. The application is installed with Data Services.

- If you use a version of the software that automatically installs the NCOALink Utility, see the following sections for information about how to locate and run the application.
- If you use a version of the software that does not install the NCOALink Utility, you can download the application from the SAP Support Portal at <https://support.sap.com/software/address-directories.html>.

NCOALink licensing requirements

USPS licensing requires that you use the most current NCOALink directories available (either weekly or monthly depending on the license type).

USPS licensing requires limited or full service provider licensees to download the daily delete file and copy it to the location where the NCOALink directories are located every day that NCOALink jobs are run.

NCOALink directories expire after 45 days.

8.5.13.1.1.5.2 Running the NCOALink Utility

Prerequisites:

Ensure that your system meets the following minimum requirements:

- At least 60 GB of available disk space
 - Sufficient RAM
1. Run the NCOALink Utility from the location where it is installed.

If your NCOALink Utility was installed with Data Services, the utility is located in the following folder, where `$LINK_DIR` is the path to your installation directory:

- `$LINK_DIR\bin\ncoa\ncoutil.exe` (Windows)

- `$LINK_DIR/bin/ncoa/ncoautl` (UNIX)

2. Use the `ncoautl` command with the following command-line options:

Option		Description
Windows	UNIX	
<code>/p:t</code>	<code>-p:t</code>	Perform transfer to copy files from the source to the destination. When using this option you must also specify the following: <ul style="list-style-type: none"> ◦ Location of compressed NCOALink data files with <code>/d</code> or <code>-d</code> ◦ Transfer destination location with <code>/t</code> or <code>-t</code>
<code>/p:u</code>	<code>-p:u</code>	Perform unpack to uncompress the files. When using this option, you must also specify the following: <ul style="list-style-type: none"> ◦ Transfer destination location with the <code>/d</code> or <code>-d</code> option ◦ Transfer destination location with <code>/t</code> or <code>-t</code>
<code>/p:v</code>	<code>-p:v</code>	Perform verification on the fields. When using this option, you must also specify the transfer destination location with <code>/t</code> or <code>-t</code> .
<code>/d</code>	<code>-d</code>	Specify location of compressed NCOALink data files.
<code>/t</code>	<code>-t</code>	Specify transfer destination location.
<code>/nos</code>	<code>-nos</code>	Do not stop on error (return failure code as exit status).
<code>/a</code>	<code>-a</code>	Answer all warning messages with Yes.

You can combine `p` options. For example, if you want to transfer, unpack, and verify all in the same process, enter `/p:tuv` or `-p:tuv`.

After performing the `p` option specified, the program closes.

❖ Example

Your command line may look something like this:

Windows

```
ncoautl /p:tuv /d D:\downloads\ncoa /t C:\Program Files (x86)\SAP
BusinessObjects\Data Services\DataQuality\reference_data
```

UNIX

```
ncoautl -a -nos -p:tuv -d /local/downloads/ncoa -t /local/dataservices/
DataQuality/reference_data
```

8.5.13.1.1.5.3 NCOALink software product information

Use the information below to complete the Compliance Testing Product Information Form. Find this form on the RIBBS website at http://ribbs.usps.gov/ncoalink/documents/tech_guides ➦. Click the Compliance Testing Form.doc link.

Compliance Testing Product Information form	Description
Company Name & License Number	Your specific information. The license number is the authorization code provided in your USPS approval letter.
Company's NCOALink Product Name	Mover ID for NCOALink
Platform or Operating System	Your specific information
NCOALink Software Vendor	SAP Americas, Inc.
NCOALink Software Product Name	Mover ID
NCOALink Software Product Version	ACE
Address Matching ZIP+4 Product Name	Contact SAP Business User Support.
Address Matching ZIP+4 Product Version	Contact SAP Business User Support.
Address Matching ZIP+4 System	Closed
Is Software Hardware Dependent?	No
DPV® Product Name	ACE
DPV Product Version	Contact SAP Business User Support.
LACSLink® Product Name	ACE
LACSLink Product Version	Contact SAP Business User Support.
NCOALink Software options: Integrated or Standalone check boxes	Integrated
ANKLink Enhancement check box (applicable for Limited Service Providers and End Users)	Check the box if you purchased the ANKLink option from SAP.
HASH—FLAT—BOTH check boxes	Indicate your preference. The software provides access to both file formats.
NCOALink Level Option check boxes	Check the appropriate box.

Related Information

[Completing NCOALink certification \[page 774\]](#)

8.5.13.1.1.5.4 Completing NCOALink certification

During certification you must process files from the USPS to prove that you adhere to the requirements of your license agreement. NCOALink certification has two stages. Stage I is an optional test which includes answers that allow you to troubleshoot and prepare for the Stage II test. The Stage II test does not contain answers and is sent to the USPS for evaluation of the accuracy of your software configuration.

Related Information

[Running the NCOALink certification jobs \[page 777\]](#)

8.5.13.1.1.5.5 NCOALink blueprints

SAP provides NCOALink blueprints. The blueprints contain the corresponding projects, jobs, data flows, and input/output formats. Additionally, the blueprints can be used to process a test file provided by the USPS during an audit.

Import NCOALink blueprints from `<DS_COMMON_DIR>\DataQuality\Certifications` where `<DS_COMMON_DIR>` is the software's common configuration location.

The following table contains the file names for the Stage I NCOALink blueprints:

Object	Name
ATL file	us_ncoalink_stage_certification.atl
Project	DataQualityCertificationNCOALink
Job	Job_DqBatchUSAReg_NCOALinkStageI
data flow	DF_DqBatchUSAReg_NCOALinkStageI
Input file format	DqUsaNCOALinkStageI_in
Output file format	DqUsaNCOALinkStageI_out

The following table contains the file names for the Stage II NCOALink blueprints:

Object	Name
ATL file	us_ncoalink_stage_certification.atl
Project	DataQualityCertificationNCOALink
Job	Job_DqBatchUSAReg_NCOALinkStageII
data flow	DF_DqBatchUSAReg_NCOALinkStageII
Input file format	DqUsaNCOALinkStageII_in
Output file format	DqUsaNCOALinkStageII_out

8.5.13.1.1.5.6 Setting up the NCOALink blueprints

Before performing the steps below you must import the NCOALink blueprints.

To set up the NCOALink Stage I and Stage II blueprints, follow the steps below.

1. In the Designer, select  **Tools** . The *Substitution Parameter Editor* opens.

2. Choose the applicable configuration from the [Default Configuration](#) drop list and enter values for your company's information and reference file locations. Click **OK** to close the Substitution Parameter Configurations tool.
3. Open the DataQualityCertificationsNCOALink project (which was imported with the blueprints).
4. Open the Job_DqBatchUSAReg_NCOALinkStagel job and then open the DF_DqBatchUSAReg_NCOALinkStagel data flow.
5. Click the DqUsaNCOALinkStagel_in file to open the [Source File Editor](#). Find the [Data Files\(s\)](#) property group in the lower portion of the editor and make the following changes:
 - a. In the [Root Directory](#) option, type the path or browse to the directory containing the input file.
If you type the path, do not type a backslash (\) or forward slash (/) at the end of the file path.
 - b. In the [File name\(s\)](#) option, change Stagel.in to the name of the Stage file provided by the USPS.
 - c. Exit the [Source File Editor](#).
6. Click the DqUsaNCOALinkStagel_out file to open the [Target File Editor](#). In the [Data Files\(s\)](#) property group make the following changes:
 - a. In the [Root Directory](#) option, type the path or browse to the directory containing the output file.
If you type the path, do not type a backslash (\) or forward slash (/) at the end of the file path.
 - b. (Optional.) In the [File name\(s\)](#) option, change StageI.out to conform to your company's file naming convention.
 - c. Exit the [Target File Editor](#).
7. Double-click the USARegulatoryNCOALink_AddressCleanse transform to open the Transform Editor and click the [Options](#) tab.
8. Enter the correct path location to the reference files in the Reference Files group as necessary. Use the \$RefFilesAddressCleanse substitution variable to save time.
9. In the USPS License Information group, do the following:
 - a. Enter a meaningful number in the [List ID](#) option.
 - b. Enter the current date in the [List Received Date](#) and [List Return Date](#) options.
 - c. Ensure that the provider level specified in the substitution parameter configuration by the \$USPSProviderLevel is accurate or specify the appropriate level (Full Service Provider, Limited Service Provider, or End User) in the [Provider Level](#) option.
 - d. If you are a full service provider or limited service provider, complete the options in the [NCOALink > PAF Details >](#) group and the [NCOALink > Service Provider Options >](#) group.
10. When you are satisfied with the results of the Stage I test, repeat steps 4 through 9 to set up the Stage II objects found in the DF_DqBatchUSAReg_NCOALinkStage II data flow.

Related Information

[Reference Guide: USPS license information options \[page 716\]](#)

[Importing certification blueprints \[page 768\]](#)

[DSF2 Certification blueprints \[page 778\]](#)

[NCOALink blueprints \[page 775\]](#)

[CASS self-certification blueprint \[page 770\]](#)

8.5.13.1.1.5.7 Running the NCOALink certification jobs

Before you run the NCOALink certification jobs, make sure that you have installed the DPV, LACSLink, and U.S. National directory files to the locations you specified during configuration and that the NCOALink directories provided by the USPS are available.

Running the Stage I job is optional; the results do not need to be sent to the USPS. However, running the Stage I job can help you ensure that you have configured the software correctly and are prepared to execute the Stage II job.

1. Use the NCOALink Utility to install the NCOALink directories provided by the USPS.
2. Download the current version of the USPS daily delete file from <https://epf.usps.gov/> .
3. Download the Stage I file from <http://ribbs.usps.gov/> and uncompress it to the location you specified when setting up the certification job.

Ensure the input file name in the source transform matches the name of the Stage I file from the USPS.

4. Execute the Stage I job and compare the test data with the expected results provided by the USPS in the Stage I input file.

As necessary, make modifications to your configuration until you are satisfied with the results of your Stage I test.

5. Download the Stage II file from the location specified by the USPS and uncompress it to the location you specified when setting up the certification job.

Ensure the input file name in the transform matches the name of the Stage II file from the USPS.

6. Execute the Stage II job. Follow the specific instructions in the *NCOALink Certification/Audit Instructions* document that the USPS should have provided to you.
7. Compress the following results (using WinZip for example) and name it using the same name as the downloaded Stage II file (step 5):
 - Stage II output file
 - NCOALink Processing Summary Report
 - CASS Form 3553
 - All log files generated in the \$\$CertificationLog path
 - Customer Service Log
 - PAF (Service Providers only)
 - Broker/Agent/List Administrator log (Service Providers only)
8. Send the results to the USPS for verification.

Related Information


[Running the NCOALink Utility \[page 772\]](#)

8.5.13.1.1.6 DSF2 Certification

DSF2 is a feature that must be purchased before testing.

The DSF2 certification consists of the following steps:

1. Application and Self-Certification Statement Approval
2. Documentation Requirements
3. Stage I Interface Development
4. DSF2 Testing and Certification
5. Execution of License


The entire process is detailed in the USPS *DSF2 Certification Package* document posted on the RIBBS website. Select the DSF2 Certification Package link on https://ribbs.usps.gov/dsf2/documents/tech_guides/ .

The *DSF2 Certification Package* contains processes and procedures and the necessary forms for you to complete the five steps listed above.

8.5.13.1.1.6.1 DSF2 Equipment Information for USPS certifications

In the DSF2 Certification Package document, there is an Equipment Information form. You are required to provide information about the software you are using to certify for DSF2. Use the information in the following table as you complete the form for the DSF2 certification process.

Equipment Information form	Description
Interface Software Vendor	SAP Americas, Inc.
Interface Software Product Name	ACE
Interface Software Product Version	Contact SAP Business User Support.
Address Matching ZIP+4 Product Name	ACE
Address Matching ZIP+4 Product Version	Contact SAP Business User Support.
Address Matching ZIP+4 System	Closed System
Interface Hardware Vendor/Model/Type	N/A The software is not hardware dependent
Interface Hardware Operating System	N/A The software is not hardware dependent
Interface Hardware Serial Number	N/A The software is not hardware dependent

Find the DSF2 Certification Package document on the RIBBS website at https://ribbs.usps.gov/dsf2/documents/tech_guides/ .

8.5.13.1.1.6.2 DSF2 Certification blueprints

SAP BusinessObjects provides DSF2 certification blueprints for the three types of DSF2 certifications. The blueprints contain the corresponding projects, jobs, data flows, and input/output formats. Additionally, the blueprints can be used to process a test file provided by the USPS during an audit.

Import the DSF2 certification blueprints from `<DS_COMMON_DIR>\DataQuality\Certifications`, where `<DS_COMMON_DIR>` is the software's common configuration location.

The following table contains the file names for the USPS DSF2 Augment certification:

Object	Name
ATL file	us_dsf2_certification.atl
Project	DataQualityCertificationDSF2
Job	Job_DqBatchUSAREg_DSF2Augment
data flow	DF_DqBatchUSAREg_DSF2Augment
Input file format	DqUsaDSF2Augment_in
Output file format	DqUsaDSF2Augment_out

The following table contains the file names for the USPS DSF2 Invoice certification:

Project	Name
ATL file	us_dsf2_certification.atl
Project	DataQualityCertificationDSF2
Job	Job_DqBatchUSAREg_DSF2Invoice
data flow	DF_DqBatchUSAREg_DSF2Invoice
Input file format	DqUsaDSF2Invoice_in
Output file format	DqUsaDSF2Invoice_out

The following table contains the file names for the USPS DSF2 Sequence certification:

Project	Name
ATL file	us_dsf2_certification.atl
Project	DataQualityCertificationDSF2
Job	Job_DqBatchUSAREg_DSF2Sequence
data flow	DF_DqBatchUSAREg_DSF2Sequence
Input file format	DqUsaDSF2Sequence_in
Output file format	DqUsaDSF2Sequence_out

8.5.13.1.1.6.3 Setting up the DSF2 certification blueprints

Before performing the steps below you must import the DSF2 blueprints.

Follow these steps to set up the DSF2 Augment, Invoice, and Sequence certification blueprints.

1. In the Designer, select **Tools** > *Substitution Parameter Configurations* .
The *Substitution Parameter Editor* opens.
2. Choose the applicable configuration from the *Default Configuration* drop list and enter values for your company's information and reference file locations.

i Note

DSF2 Augment only. Remember to enter the static directories location for the \$\$RefFilesUSPSStatic substitution variable.

3. Open the DataQualityCertificationDSF2 project (downloaded with the blueprint).
4. Expand the desired certification job and data flow. For example, if you are setting up for DSF2 Augment, expand the Job_DqBatchUSAReg_DSF2Augment job and then the DF_DqBatchUSAReg_DSF2Augment data flow.
5. Double-click the applicable input file format (*.in) to open the [Source File Editor](#). For example, for DSF2 Augment certification, double-click DSF2_Augment.in.
6. In the [Data Files\(s\)](#) property group make the following changes:
 - a. In the [Root Directory](#) option, type the path or browse to the directory containing the input file.
If you type the path, do not type a backslash or forward slash at the end of the file path.
 - b. In the [File name\(s\)](#) option, change the input file name to the name of the file provided by the USPS.
7. Double-click the applicable output file format (*.out) to open the Target File Editor. For example, for DSF2 Augment certification, double-click DSF2_Augment.out.
8. In the Data Files(s) property group make the following changes:
 - a. In the [Root Directory](#) option, type the path or browse to the directory containing the output file.
If you type the path, do not type a backslash or forward slash at the end of the file path.
 - b. (Optional) In the [File name\(s\)](#) option, change the output file name to conform to your company's file naming convention.
9. Click the USARegulatory_AddressCleanse transform to open the Transform Editor and click the [Options](#) tab.

i Note

For DSF2 Sequence and Invoice certifications, you will open the DSF2_Walk_Sequencer transform.

10. As necessary, in the Reference Files group, enter the correct path location to the reference files.
For DSF2 Augment certification, use the \$\$RefFilesUSPSStatic substitution variable to save time.
11. In the CASS Report Options, update each option that is listed as "CHANGE_THIS" if applicable.

Related Information

[Importing certification blueprints \[page 768\]](#)

[DSF2 Certification blueprints \[page 778\]](#)

[NCOALink blueprints \[page 775\]](#)

[CASS self-certification blueprint \[page 770\]](#)

8.5.13.2 Country coverage

This table contains information about address cleansing and geocode coverage per country.

The “Country Reference Data” column identifies which country-specific reference data covers each country for address cleansing. For each country that is covered by a country-specific reference data the chart identifies which languages and scripts are supported in the reference data, and to what depth address data can be validated.

The validation levels from poorest to finest are as follows.

- City – Validation is to the city (locality) and postcode.
- Primary – Validation is to the street, and sometimes to the house number.
- Secondary – Validation is to the building or secondary data such as floor, apartment, or suite.

In the table below, countries where the Validation Level column is blank use the All-World reference data. This data supports the local language in Latin script and validates to the city level.

Validation to a particular level means that when the address data sent in the request matches data in the reference data, then errors can be corrected and missing components can be completed. While the remaining address data is not validated, it may still be standardized and formatted to the country norms and the address cleansing rules.

The “Geo” column identifies which countries are supported for returning geo-location coordinates.

i Note

Geo and Reverse Geo are not supported for US Territories.

Country Code	Country Name	Country Reference Data	Languages	Scripts	Validation Level	Geo
AD	Andorra	Spain	Spanish	Latin	Primary	
AE	United Arab Emirates					
AF	Afghanistan					
AG	Antigua and Barbuda					
AI	Anguilla					
AL	Albania					
AM	Armenia					
AO	Angola					
AQ	Antarctica				N/A for All World directories Country Identification Engine will recognize country	
AR	Argentina					
AS	American Samoa	United States	English	Latin	Primary	

Country Code	Country Name	Country Reference Data	Languages	Scripts	Validation Level	Geo
AT	Austria	Austria	German	Latin	Secondary	Yes
AU	Australia	Australia	English	Latin	Primary	Yes
AW	Aruba					
AX	Åland Islands	Finland	Swedish and Finnish	Latin	City	
AZ	Azerbaijan					
BA	Bosnia and Herzegovina					
BB	Barbados					
BD	Bangladesh					
BE	Belgium	Belgium	Dutch, French, German	Latin	Primary	Yes
BF	Burkina Faso					
BG	Bulgaria				City	
BH	Bahrain					
BI	Burundi					
BJ	Benin					
BL	Saint Barthélemy	France	French	Latin	Secondary	
BM	Bermuda					
BN	Brunei Darussalam					
BO	Plurinational State of Bolivia					
BQ	Bonaire, Sint Eustatius and Saba					
BS	Bahamas (the)					
BT	Bhutan					
BV	Bouvet Island				N/A for All World directories Country Identification Engine will recognize country	
BW	Botswana					
BY	Belarus					
BZ	Belize					
CA	Canada	Canada	English, French	Latin	Secondary	Yes
CC	Cocos (Keeling) Islands	Australia	English	Latin	Primary	

Country Code	Country Name	Country Reference Data	Languages	Scripts	Validation Level	Geo
CD	Congo (Democratic Republic of the)					
CF	Central African Republic (the)					
CG	Republic of Congo					
CH	Switzerland	Switzerland	German, French, Italian	Latin	Secondary	Yes
CI	Côte d'Ivoire					
CK	Cook Islands (the)					
CL	Chile					
CM	Cameroon					
CN	People's Republic of China	China	Chinese	Simplified Chinese, Latin	Primary	
CO	Colombia					
CR	Costa Rica					
CU	Cuba					
CV	Cabo Verde					
CW	Curaçao					
CX	Christmas Island	Australia	English	Latin	Primary	
CY	Cyprus					
CZ	Czechia	Czechia	Czech	Latin	Secondary Locality aliases for English, Italian, and German	Yes
DE	Germany	Germany	German	Latin	Primary	Yes
DJ	Djibouti					
DK	Denmark	Denmark	Danish	Latin	Primary	Yes
DM	Dominica					
DO	Dominican Republic (the)					
DZ	Algeria					
EC	Ecuador					
EE	Estonia	Estonia	Estonian	Latin	Secondary	Yes
EG	Egypt					

Country Code	Country Name	Country Reference Data	Languages	Scripts	Validation Level	Geo
EH	Western Sahara				N/A for All World directories Country Identification Engine will recognize country	
ER	Eritrea					
ES	Spain	Spain	Spanish (including Catalan, Galician, and Basque)	Latin	Primary	Yes
ET	Ethiopia					
FI	Finland	Finland	Finnish Swedish	Latin	Primary	Yes
FJ	Fiji					
FK	Falkland Islands (the) (Malvinas)					
FM	Micronesia (Federated States of)	United States	English	Latin	Primary	
FO	Faroe Islands (the)	Denmark	Danish	Latin	Primary	
FR	France	France	French	Latin	Secondary	Yes
GA	Gabon					
GB	United Kingdom of Great Britain and Northern Ireland	United Kingdom	English	Latin	Secondary	Yes
GD	Grenada					
GE	Georgia					
GF	French Guiana	France	French	Latin	Secondary	
GG	Guernsey	United Kingdom	English	Latin	Secondary	
GH	Ghana					
GI	Gibraltar					
GL	Greenland	Denmark	Danish, Kalaallit	Latin	City	
GM	Gambia (the)					
GN	Guinea					
GP	Guadeloupe	France	French	Latin	Secondary	

Country Code	Country Name	Country Reference Data	Languages	Scripts	Validation Level	Geo
GQ	Equatorial Guinea					
GR	Greece				Primary	
GS	South Georgia and the South Sandwich Islands				N/A for All World directories Country Identification Engine will recognize country	
GT	Guatemala					
GU	Guam	United States	English	Latin	Secondary	
GW	Guinea-Bissau					
GY	Guyana					
HK	Hong Kong					
HM	Heard Island and McDonald Islands				N/A for All World directories Country Identification Engine will recognize country	
HN	Honduras					
HR	Croatia					
HT	Haiti					
HU	Hungary	Hungary	Hungarian	Latin	Primary	
ID	Indonesia					
IE	Ireland					
IL	Israel					
IM	Isle of Man	United Kingdom	English	Latin	Secondary	
IN	India	India	English	Latin	Primary	
IO	British Indian Ocean Territory (the)				N/A for All World directories Country Identification Engine will recognize country	

Country Code	Country Name	Country Reference Data	Languages	Scripts	Validation Level	Geo
IQ	Iraq					
IR	Iran (Islamic Republic of)					
IS	Iceland					
IT	Italy	Italy	Italian	Latin	Primary	Yes
JE	Jersey	United Kingdom	English	Latin	Secondary	
JM	Jamaica					
JO	Jordan					
JP	Japan	Japan	Japanese	Kanji, Hiragana, Katakana	Secondary	
KE	Kenya					
KG	Kyrgyzstan					
KH	Cambodia					
KI	Kiribati					
KM	Comoros (the)					
KN	Saint Kitts and Nevis					
KP	Korea (Democratic People's Republic of)					
KR	Korea (Republic of)	Korea	Korean	Hangul, Latin	Primary	
KW	Kuwait					
KY	Cayman Islands (the)					
KZ	Kazakhstan					
LA	Lao People's Democratic Republic (the)					
LB	Lebanon					
LC	Saint Lucia					
LI	Liechtenstein	Switzerland	German	Latin	Primary	Yes
LK	Sri Lanka					
LR	Liberia					
LS	Lesotho					
LT	Lithuania	Lithuania	Lithuanian	Latin	Primary	Yes
LU	Luxembourg	Luxembourg	French, German, Lëtzebuergesch	Latin	Primary	Yes
LV	Latvia	Latvia	Latvian	Latin	Secondary	Yes
LY	Libya					

Country Code	Country Name	Country Reference Data	Languages	Scripts	Validation Level	Geo
MA	Morocco					
MC	Monaco	France	French	Latin	Primary (directory contains some organization information)	
MD	Moldova (the Republic of)					
ME	Montenegro					
MF	Saint Martin (French part)	France	French	Latin	Secondary	Yes
MG	Madagascar					
MH	Marshall Islands (the)	United States	English	Latin	Primary	
MK	Macedonia (the Former Yugoslav Republic of)					
ML	Mali					
MM	Myanmar					
MN	Mongolia					
MO	Macao (also known as Macau)	Macao	Chinese	Traditional Chinese, Latin	Primary	Yes
MP	Northern Mariana Islands (the)	United States	English	Latin	Primary	
MQ	Martinique	France	French	Latin	Secondary	
MR	Mauritania					
MS	Montserrat					
MT	Malta					
MU	Mauritius					
MV	Maldives					
MW	Malawi					
MX	Mexico	Mexico	Spanish	Latin	Primary	
MY	Malaysia					
MZ	Mozambique					
NA	Namibia					
NC	New Caledonia	France	French	Latin	City	
NE	Niger (the)					
NF	Norfolk Island	Australia	English	Latin	Primary	
NG	Nigeria					
NI	Nicaragua					

Country Code	Country Name	Country Reference Data	Languages	Scripts	Validation Level	Geo
NL	Netherlands	Netherlands	Dutch	Latin	Primary	Yes
NO	Norway	Norway	Norwegian	Latin	Primary	Yes
NP	Nepal					
NR	Nauru					
NU	Niue				N/A for All World Directories Country Identification Engine will recognize country	
OM	Oman					
PA	Panama					
PE	Peru					
PF	French Polynesia	France	French	Latin	City	
PG	Papua New Guinea					
PH	Philippines (the)					
PK	Pakistan					
PL	Poland	Poland	Polish	Latin	Primary	Yes
PM	Saint Pierre and Miquelon	France	French	Latin	City	
PN	Pitcairn					
PR	Puerto Rico	United States	Spanish	Latin	Secondary	
PS	Palestine (State of)					
PT	Portugal	Portugal	Portuguese	Latin	Secondary	Yes
PW	Palau	United States	English	Latin	Primary	
PY	Paraguay					
QA	Qatar					
RE	Réunion	France	French	Latin	Secondary	
RO	Romania					
RS	Serbia					
RU	Russian Federation (the)	Russia	Russian	Cyrillic, Latin	Secondary	
RW	Rwanda					
SA	Saudi Arabia					
SB	Solomon Islands					
SC	Seychelles					

Country Code	Country Name	Country Reference Data	Languages	Scripts	Validation Level	Geo
SD	Sudan (the)					
SE	Sweden	Sweden	Swedish	Latin	Primary	Yes
SG	Singapore	Singapore	English	Latin	Secondary	Yes
SH	Saint Helena, Ascension and Tristan da Cunha					
SI	Slovenia					
SJ	Svalbard and Jan Mayen	Norway	Norwegian	Latin	Primary	
SK	Slovakia	Slovakia	Slovakian	Latin	Primary	
SL	Sierra Leone					
SM	San Marino	Italy	Italian	Latin	Primary	
SN	Senegal					
SO	Somalia					
SR	Suriname					
SS	South Sudan					
ST	Sao Tome and Principe					
SV	El Salvador					
SX	Sint Maarten (Dutch part)					
SY	Syrian Arab Republic					
SZ	Swaziland					
TC	Turks and Caicos Islands (the)					
TD	Chad					
TF	French Southern Territories (the)					
TG	Togo					
TH	Thailand					
TJ	Tajikistan					
TK	Tokelau				N/A for All World Directories Country Identification Engine will recognize country	

Country Code	Country Name	Country Reference Data	Languages	Scripts	Validation Level	Geo
TL	Timor-Leste				N/A for All World Directories Country Identification Engine will recognize country	
TM	Turkmenistan					
TN	Tunisia					
TO	Tonga					
TR	Turkey	Turkey	Turkish	Latin	Primary	Yes
TT	Trinidad and Tobago					
TV	Tuvalu					
TW	Taiwan (Province of China)	Taiwan	Chinese	Traditional Chinese, Latin	Primary Supported with the All World directory	Yes
TZ	Tanzania, United Republic of					
UA	Ukraine					
UG	Uganda					
UM	United States Minor Outlying Islands (the)				N/A for All World Directories Country Identification Engine will recognize country	
US	United States of America (the)	United States	English	Latin	Secondary	Yes
UY	Uruguay					
UZ	Uzbekistan					
VA	Holy See (the)	Italy	Italian	Latin	Primary	
VC	Saint Vincent and the Grenadines					
VE	Venezuela (Bolivarian Republic of)					
VG	Virgin Islands (British)					
VI	Virgin Islands (U.S.)	United States	English	Latin	Secondary	Yes

Country Code	Country Name	Country Reference Data	Languages	Scripts	Validation Level	Geo
VN	Viet Nam					
VU	Vanuatu					
WF	Wallis and Futuna	France	French	Latin	City	
WS	Samoa					
YE	Yemen					
YT	Mayotte	France	French	Latin	Secondary	
ZA	South Africa					
ZM	Zambia					
ZW	Zimbabwe					

8.5.13.3 Country ISO codes and assignment engines

The table shows which engine (if any) provides address correction. Additionally, it lists the 2-character and 3-character ISO code, the 3-digit ISO code, European Postcode prefix, and the level of assignment. The assignment level is based on the reference data that you own.

Table Key

Engine	Assignment Level
Canada = C	Country = C
Global Address = G	Locality = L
USA = U	Primary Name = Pn
	Premise = Pr
	Secondary = S

Country name	2-char ISO code	3-char ISO code	3-digit ISO code	European Post-code prefix	Engine	Assignment level
Afghanistan	AF	AFG	004		G	C, L
Åland Islands	AX	ALA	248	AX	G	C, L
Albania	AL	ALB	008		G	C, L
Algeria	DZ	DZA	012		G	C, L
American Samoa	AS	ASG	016		U	C, L, Pn, Pr, S
					G	C, L
Andorra	AD	AND	020	AND	G	C, L
Angola	AO	AGO	024		G	C, L

Country name	2-char ISO code	3-char ISO code	3-digit ISO code	European Post-code prefix	Engine	Assignment level
Anguilla	AI	AIA	660		G	C, L
Antarctica	AQ	ATA	010		G	C
Antigua and Barbuda	AG	ATG	028		G	C, L
Argentina	AR	ARG	032		G	C, L
Armenia	AM	ARM	051		G	C, L
Aruba	AW	ABW	533		G	C, L
Australia	AU	AUS	036		G	C, L, Pn, Pr, S
Austria	AT	AUT	040	A	G	C, L, Pn, Pr, S
Azerbaijan	AZ	AZE	031		G	C, L
Bahamas	BS	BHS	044		G	C, L
Bahrain	BH	BHR	048		G	C, L
Bangladesh	BD	BGD	050		G	C, L
Barbados	BB	BRB	052		G	C, L
Belarus	BY	BLR	112		G	C, L
Belgium	BE	BEL	056	B	G	C, L, Pn, Pr
Belize	BZ	BLZ	084		G	C, L
Benin	BJ	BEN	204		G	C, L
Bermuda	BM	BMU	060		G	C, L
Bhutan	BT	BTN	064		G	C, L
Bolivia, Plurinational State of	BO	BOL	068		G	C, L
Bonaire, Sint Eustatius, and Saba	BQ	BES	535		G	C, L
Bosnia and Herzegovina	BA	BIH	070		G	C, L
Botswana	BW	BWA	072		G	C, L
Bouvet Island	BV	BVT	074		G	C
Brazil	BR	BRA	076		G	C, L, Pn, Pr
British Indian Ocean Territory	IO	IOT	086		G	C
British Virgin Islands	VG	VGB	092		G	C, L
Brunei Darussalam	BN	BRN	096		G	C, L
Bulgaria	BG	BGR	100	BG	G	C, L

Country name	2-char ISO code	3-char ISO code	3-digit ISO code	European Post-code prefix	Engine	Assignment level
Burkina Faso	BF	BFA	854		G	C, L
Burundi	BI	BDI	108		G	C, L
Cambodia	KH	KHM	116		G	C, L
Cameroon	CM	CMR	120		G	C, L
Canada	CA	CAN	124		C	C, L, Pn, Pr, S
					G	C, L
Cape Verde	CV	CPV	132		G	C
Cayman Islands	KY	CYM	136		G	C
Central African Republic	CF	CAF	140		G	C, L
Chad	TD	TCD	148		G	C, L
Chile	CL	CHL	152		G	C, L
China	CN	CHN	156		G	C, L, Pn, Pr
Christmas Island (Included in the Australia data package)	CX	CXR	162		G	C, L
Cocos (Keeling) Isles (Included in the Australia data package)	CC	CCK	166		G	C, L
Colombia	CO	COL	170		G	C, L
Comoros	KM	COM	174		G	C, L
Congo, Democratic Republic of	CD	COD	180		G	C, L
Congo, Republic of	CG	COG	178		G	C, L
Cook Islands	CK	COK	184		G	C, L
Costa Rica	CR	CRI	188		G	C, L
Côte d'Ivoire (Ivory Coast)	CI	CIV	384		G	C, L
Croatia (Hrvatska)	HR	HRV	191	HR	G	C, L
Cuba	CU	CUB	192		G	C, L
Curaçao	CW	CUW	531		G	C, L
Cyprus	CY	CYP	196	CY	G	C, L

Country name	2-char ISO code	3-char ISO code	3-digit ISO code	European Post-code prefix	Engine	Assignment level
Czechia	CZ	CZE	203	CZ	G	C, L, Pn, Pr
Democratic People's Republic of Korea	KP	PRK	408		G	C, L
Denmark	DK	DNK	208	DK	G	C, L, Pn, Pr
Djibouti	DJ	DJI	262		G	C, L
Dominica	DM	DMA	212		G	C, L
Dominican Republic	DO	DOM	214		G	C, L
Ecuador	EC	ECU	218		G	C, L
Egypt	EG	EGY	818		G	C, L
El Salvador	SV	SLV	222		G	C, L
Equatorial Guinea	GQ	GNQ	226		G	C, L
Eritrea	ER	ERI	232		G	C, L
Estonia	EE	EST	233	EE	G	C, L, Pn, Pr
Ethiopia	ET	ETH	231		G	C, L
Falkland Islands	FK	FLK	238		G	C, L
Faroe Islands (Included in the Denmark data package)	FO	FRO	234	FO	G	C, L, Pn, Pr
Federated States of Micronesia	FM	FSM	583		U G	C, L, Pn, Pr, S C, L
Fiji	FJ	FJI	242		G	C, L
Finland	FI	FIN	246	FI	G	C, L, Pn, Pr
France	FR	FRA	250	F	G	C, L, Pn, Pr, S
French Guiana (Included in the France data package)	GF	GUF	254		G	C, L, Pn, Pr
French Polynesia (Included in the France data package)	PF	PYF	258		G	C, L, Pn, Pr
French Southern Territories	TF	ATF	260		G	C, L, Pn, Pr
Gabon	GA	GAB	266		G	C, L
Gambia	GM	GMB	270		G	C, L

Country name	2-char ISO code	3-char ISO code	3-digit ISO code	European Post-code prefix	Engine	Assignment level
Georgia	GE	GEO	268		G	C, L
Germany	DE	DEU	276	D	G	C, L, Pn, Pr
Ghana	GH	GHA	288		G	C, L
Gibraltar	GI	GIB	292		G	C, L
Greece	GR	GRC	300	GR	G	C, L, Pn, Pr
Greenland (Included in the Denmark data package)	GL	GRL	304	GL	G	C, L, Pn, Pr
Grenada	GD	GRD	308		G	C, L
Guadeloupe (Included in the France data package)	GP	GLP	312		G	C, L, Pn, Pr
Guam	GU	GUM	316		U G	C, L, Pn, Pr, S C, L
Guatemala	GT	GTM	320		G	C, L
Guernsey (Included in the United Kingdom data package)	GG	GGY	831	G	G	C, L, Pn, Pr, S
Guinea	GN	GIN	324		G	C, L
Guinea-Bissau	GW	GNB	624		G	C, L
Guyana	GY	GUY	328		G	C, L
Haiti	HT	HTI	332		G	C, L
Heard Island and McDonald Islands	HM	HMD	334		G	C, L
Holy See (Vatican City State) (Included in the Italy data package)	VA	VAT	336		G	C, L, Pn, Pr
Honduras	HN	HND	340		G	C, L
Hong Kong	HK	HKG	344		G	C, L
Hungary	HU	HUN	348	H	G	C, L, Pn, Pr
Iceland	IS	ISL	352	IS	G	C, L
India	IN	IND	356		G	C, L, Pn, Pr
Indonesia	ID	IDN	360		G	C, L

Country name	2-char ISO code	3-char ISO code	3-digit ISO code	European Post-code prefix	Engine	Assignment level
Iraq	IQ	IRQ	368		G	C, L
Ireland, Republic of	IE	IRL	372	IRL	G	C, L
Islamic Republic of Iran	IR	IRN	364		G	C, L
Isle of Man (Included in the United Kingdom data package)	IM	IMN	833		G	C, L, Pn, Pr, S
Israel	IL	ISR	376		G	C, L
Italy	IT	ITA	380	I	G	C, L, Pn, Pr
Jamaica	JM	JAM	388		G	C, L
Japan	JP	JPN	392		G	C, L, Pn, Pr, S
Jersey (Included in the United Kingdom data package)	JE	JEY	832		G	C, L, Pn, Pr, S
Jordan	JO	JOR	400		G	C, L
Kazakhstan	KZ	KAZ	398		G	C, L
Kenya	KE	KEN	404		G	C, L
Kiribati	KI	KIR	296		G	C, L
Kuwait	KW	KWT	414		G	C, L
Kyrgyzstan	KG	KGZ	417		G	C, L
Lao People's Democratic Republic	LA	LAO	418		G	C, L
Latvia	LV	LVA	428	LV	G	C, L, Pn, Pr
Lebanon	LB	LBN	422		G	C, L
Lesotho	LS	LSO	426		G	C, L
Liberia	LR	LBR	430		G	C, L
Libya	LY	LBY	434		G	C, L
Liechtenstein (Included in the Switzerland data package)	LI	LIE	438	FL	G	C, L, Pn, Pr
Lithuania	LT	LTU	440	LT	G	C, L, Pn, Pr
Luxembourg	LU	LUX	442	L	G	C, L, Pn, Pr

Country name	2-char ISO code	3-char ISO code	3-digit ISO code	European Post-code prefix	Engine	Assignment level
Macao	MO	MAC	446		G	C, L
Macedonia	MK	MKD	807	MK	G	C, L
Madagascar	MG	MDG	450		G	C, L
Malawi	MW	MWI	454		G	C, L
Malaysia	MY	MYS	458	M	G	C,L
Maldives	MV	MDV	462		G	C, L
Mali	ML	MLI	466		G	C, L
Malta	MT	MLT	470		G	C, L
Marshall Is-lands	MH	MHL	584		U	C, L, Pn, Pr, S
					G	C, L
Martinique (Included in the France data package)	MQ	MTQ	474		G	C, L, Pn, Pr
Mauritania	MR	MRT	478		G	C, L
Mauritius	MU	MUS	480		G	C, L
Mayotte (Included in the France data package)	YT	MYT	175		G	C, L, Pn, Pr
Mexico	MX	MEX	484		G	C, L
Moldova	MD	MDA	498	MD	G	C, L
Monaco (Included in the France data package)	MC	MCO	492	F	G	C, L, Pn, Pr
Mongolia	MN	MNG	496		G	C, L
Montenegro	ME	MNE	499		G	C, L
Montserrat	MS	MSR	500		G	C, L
Morocco	MA	MAR	504		G	C, L
Mozambique	MZ	MOZ	508		G	C, L
Myanmar	MM	MMR	104		G	C, L
Namibia	NA	NAM	516		G	C, L
Nauru	NR	NRU	520		G	C, L
Nepal	NP	NPL	524		G	C, L
Netherlands	NL	NLD	528	NL	G	C, L, Pn, Pr

Country name	2-char ISO code	3-char ISO code	3-digit ISO code	European Post-code prefix	Engine	Assignment level
New Caledonia (Included in the France data package)	NC	NCL	540		G	C, L, Pn, Pr
New Zealand	NZ	NZL	554		G	C, L, Pn, Pr, S
Nicaragua	NI	NIC	558		G	C, L
Niger	NE	NER	562		G	C, L
Nigeria	NG	NGA	566		G	C, L
Niue	NU	NIU	570		G	C, L
Norfolk Island (Included in the Australia data package)	NF	NFK	574		G	C, L
Northern Mariana Islands	MP	MNP	580		U	C, L, Pn, Pr, S
					G	C, L
Norway	NO	NOR	578	N	G	C, L, Pn, Pr
Occupied Palestinian Territory	PS	PSE	275		G	C
Oman	OM	OMN	512		G	C, L
Pakistan	PK	PAK	586		G	C, L
Palau	PW	PLW	585		U	C, L, Pn, Pr, S
					G	C, L
Panama	PA	PAN	591		G	C, L
Papua New Guinea	PG	PNG	598		G	C, L
Paraguay	PY	PRY	600		G	C, L
Peru	PE	PER	604		G	C, L
Philippines	PH	PHL	608		G	C, L
Pitcairn	PN	PCN	612		G	C, L
Poland	PL	POL	616	PL	G	C, L, Pn, Pr
Portugal	PT	PRT	620	P	G	C, L, Pn, Pr, S
Province of China Taiwan	TW	TWN	158		G	C, L
Puerto Rico	PR	PRI	630		U	C, L, Pn, Pr, S
					G	C, L
Qatar	QA	QAT	634		G	C, L

Country name	2-char ISO code	3-char ISO code	3-digit ISO code	European Post-code prefix	Engine	Assignment level
Republic of Korea	KR	KOR	410		G	C, L
Réunion (Included in the France data package)	RE	REU	638		G	C, L, Pn, Pr
Romania	RO	ROU	642	RO	G	C, L
Russian Federation	RU	RUS	643	RUS	G	C, L
Rwanda	RW	RWA	646		G	C, L
Saint Barthélemy (Included in the France data package)	BL	BLM	652		G	C, L
Saint Helena, Ascension, and Tristan da Cunha	SH	SHN	654		G	C, L
Saint Kitts and Nevis	KN	KNA	659		G	C, L
Saint Lucia	LC	LCA	662		G	C, L
Saint Martin (French part) (Included in the France data package)	MF	MAF	663		G	C, L
Saint Pierre and Miquelon (Included in the France data package)	PM	SPM	666		G	C, L, Pn, Pr
Saint Vincent & Grenadines	VC	VCT	670		G	C, L
Samoa	WS	WSM	882		G	C, L
San Marino (Included in the Italy data package)	SM	SMR	674	SMR	G	C, L, Pn, Pr
Sao Tome and Principe	ST	STP	678		G	C, L
Saudi Arabia	SA	SAU	682		G	C, L
Senegal	SN	SEN	686		G	C, L
Serbia	RS	SRB	688		G	C, L

Country name	2-char ISO code	3-char ISO code	3-digit ISO code	European Post-code prefix	Engine	Assignment level
Seychelles	SC	SYC	690		G	C, L
Sierra Leone	SL	SLE	694		G	C, L
Singapore	SG	SGP	702		G	C, L
Sint Maarten (Dutch Part)	SX	SXM	534		G	C, L
Slovakia	SK	SVK	703		G	C, L, Pn, Pr
Slovenia	SI	SVN	705		G	C, L
Solomon Islands	SB	SLB	090		G	C, L
Somalia	SO	SOM	706		G	C, L
South Africa	ZA	ZAF	710		G	C, L
South Georgia and the South Sandwich Islands	GS	SGS	239		G	C, L
South Sudan	SS	SDN	728		G	C, L
Spain	ES	ESP	724	E	G	C, L, Pn, Pr
Sri Lanka	LK	LKA	144		G	C, L
Sudan	SD	SDN	736		G	C, L
Suriname	SR	SUR	740		G	C, L
Svalbard and Jan Mayen (Included in the Norway data package)	SJ	SJM	744		G	C
Swaziland	SZ	SWZ	748		G	C, L
Sweden	SE	SWE	752	S	G	C, L, Pn, Pr
Switzerland	CH	CHE	756	CH	G	C, L, Pn, Pr
Syrian Arab Republic	SY	SYR	760		G	C, L
Tajikistan	TJ	TJK	762		G	C, L
Thailand	TH	THA	764		G	C, L
Timor-Leste	TL	TLS	626		G	C
Togo	TG	TGO	768		G	C, L
Tokelau	TK	TKL	772		G	C, L
Tonga	TO	TON	776		G	C, L
Trinidad and Tobago	TT	TTO	780		G	C, L

Country name	2-char ISO code	3-char ISO code	3-digit ISO code	European Post-code prefix	Engine	Assignment level
Tunisia	TN	TUN	788	TN	G	C, L
Turkey	TR	TUR	792	TR	G	C, L, Pn, Pr
Turkmenistan	TM	TKM	795		G	C, L
Turks and Cai- cos Islands	TC	TCA	796		G	C, L
Tuvalu	TV	TUV	798		G	C, L
U.S. Virgin Is- lands	VI	VIR	850		U G	C, L, Pn, Pr, S
Uganda	UG	UGA	800		G	C, L
Ukraine	UA	UKR	804	UK	G	C, L
United Arab Emirates	AE	ARE	784		G	C, L
United King- dom	GB	GBR	826	GB	G	C, L, Pn, Pr, S
United Republic of Tanzania	TZ	TZA	834		G	C, L
United States	US	USA	840		U G	C, L, Pn, Pr, S C, L
United States Minor Outlying Islands	UM	UMI	581		U G	C, L, Pn, Pr, S
Uruguay	UY	URY	858		G	C, L
Uzbekistan	UZ	UZB	860		G	C, L
Vanuatu	VU	VUT	548		G	C, L
Venezuela	VE	VEN	862		G	C, L
Viet Nam	VN	VNM	704		G	C, L
Wallis and Fu- tuna	WF	WLF	876		G	C, L, Pn, Pr
Western Sahara	EH	ESH	732		G	C, L
Yemen	YE	YEM	887		G	C, L
Zambia	ZM	ZMB	894		G	C, L
Zimbabwe	ZW	ZWE	716		G	C, L

8.5.13.4 Countries supported by lastline drilldown

The Enable Lastline Drilldown option requires a country on input. You can use either the English country name or the two-character ISO country code as input, or for some countries the country name in its native script. When the input country name is in its native script, the drilldown list will also be provided in the native script.

The Enable Lastline Drilldown option is supported by the Global Engine engine of the Global Address transform; it is not available for the Canada and USA engines.

Country name	Two-character ISO code	Country name in native script
Afghanistan	AF	
Åland Islands	AX	
Albania	AL	
Algeria	DZ	
American Samoa	AS	
Andorra	AD	
Angola	AO	
Antigua and Barbuda	AG	
Argentina	AR	
Armenia	AM	
Australia	AU	
Austria	AT	
Azerbaijan	AZ	
Bahamas	BS	
Bahrain	BH	
Bangladesh	BD	
Barbados	BB	
Belarus	BY	
Belgium	BE	
Belize	BZ	
Benin	BJ	
Bermuda	BM	
Bhutan	BT	
Bolivarian Republic of Venezuela	VE	
Bonaire, Sint Eustatius, and Saba	BQ	
Botswana	BW	
Brazil	BR	
British Virgin Islands	VG	

Country name	Two-character ISO code	Country name in native script
Brunei Darussalam	BN	
Bulgaria	BG	
Burkina Faso	BF	
Burundi	BI	
CABO VERDE	CV	
Cambodia	KH	
Cameroon	CM	
Canada	CA	
Cayman Islands	KY	
Central African Republic	CF	
Chad	TD	
Chile	CL	
China	CN	中国
Christmas Island	CX	
Cocos (Keeling) Islands	CC	
Colombia	CO	
Comoros	KM	
Congo	CG	
Costa Rica	CR	
Côte d'Ivoire	CI	
Croatia	HR	
Cuba	CU	
Curaçao	CW	
Cyprus	CY	
Czechia	CZ	
Democratic People's Republic of Korea	KP	
Denmark	DK	
Djibouti	DJ	
Dominica	DM	
Dominican Republic	DO	
Ecuador	EC	
Egypt	EG	
El Salvador	SV	
Equatorial Guinea	GQ	

Country name	Two-character ISO code	Country name in native script
Eritrea	ER	
Estonia	EE	
Ethiopia	ET	
Faroe Islands	FO	
Federated States of Micronesia	FM	
Fiji	FJ	
Finland	FI	
France	FR	
French Guiana	GF	
French Polynesia	PF	
Gabon	GA	
Gambia	GM	
Georgia	GE	
Germany	DE	
Ghana	GH	
Greece	GR	ΕΛΛΑΔΑ
Greenland	GL	
Grenada	GD	
Guadeloupe	GP	
Guam	GU	
Guatemala	GT	
Guernsey	GG	
Guinea	GN	
Guinea-Bissau	GW	
Guyana	GY	
Haiti	HT	
Holy See (Vatican City State)	VA	
Honduras	HN	
Hungary	HU	
Iceland	IS	
India	IN	
Indonesia	ID	
Iraq	IQ	
Ireland	IE	

Country name	Two-character ISO code	Country name in native script
Islamic Republic of Iran	IR	
Isle of Man	IM	
Italy	IT	
Jamaica	JM	
Japan	JP	日本
Jersey	JE	
Jordan	JO	
Kazakhstan	KZ	
Kenya	KE	
Kiribati	KI	
Kyrgyzstan	KG	
Lao People's Democratic Republic	LA	
Latvia	LV	
Lebanon	LB	
Lesotho	LS	
Liberia	LR	
Libya	LY	
Liechtenstein	LI	
Lithuania	LT	
Luxembourg	LU	
Madagascar	MG	
Malawi	MW	
Malaysia	MY	
Maldives	MV	
Mali	ML	
Malta	MT	
Marshall Islands	MH	
Martinique	MQ	
Mauritania	MR	
Mauritius	MU	
Mayotte	YT	
Mexico	MX	
Monaco	MC	
Mongolia	MN	

Country name	Two-character ISO code	Country name in native script
Montenegro	ME	
Montserrat	MS	
Morocco	MA	
Mozambique	MZ	
Myanmar	MM	
Namibia	NA	
Nauru	NR	
Nepal	NP	
Netherlands	NL	
New Caledonia	NC	
Nicaragua	NI	
Niger	NE	
Nigeria	NG	
Norfolk Island	NF	
Northern Mariana Islands	MP	
Norway	NO	
Oman	OM	
Pakistan	PK	
Palau	PW	
Panama	PA	
Papua New Guinea	PG	
Paraguay	PY	
Peru	PE	
Philippines	PH	
Plurinational State of Bolivia	BO	
Poland	PL	
Portugal	PT	
Province of China Taiwan	TW	台灣
Puerto Rico	PR	
Qatar	QA	
Republic of Korea	KR	대한민국
Republic of Moldova	MD	
Réunion	RE	
Romania	RO	

Country name	Two-character ISO code	Country name in native script
Russian Federation	RU	РОССИЯ
Rwanda	RW	
Saint Barthélemy	BL	
Saint Helena, Ascension, and Tristan da Cunha	SH	
Saint Kitts and Nevis	KN	
Saint Lucia	LC	
Saint Martin (French part)	MF	
Saint Pierre and Miquelon	PM	
Saint Vincent and the Grenadines	VC	
Samoa	WS	
San Marino	SM	
Sao Tome and Principe	ST	
Saudi Arabia	SA	
Senegal	SN	
Seychelles	SC	
Sierra Leone	SL	
Sint Maarten (Dutch Part)	SX	
Slovakia	SK	
Slovenia	SI	
Solomon Islands	SB	
Somalia	SO	
South Africa	ZA	
South Sudan	SS	
Spain	ES	
Sri Lanka	LK	
Sudan	SD	
Suriname	SR	
Svalbard and Jan Mayen	SJ	
Swaziland	SZ	
Sweden	SE	
Switzerland	CH	
Syrian Arab Republic	SY	
Tajikistan	TJ	
Thailand	TH	

Country name	Two-character ISO code	Country name in native script
The Democratic Republic of the Congo	CD	
Togo	TG	
Tonga	TO	
Trinidad and Tobago	TT	
Tunisia	TN	
Turkey	TR	
Turkmenistan	TM	
Turks and Caicos Islands	TC	
Tuvalu	TV	
U.S. Virgin Islands	VI	
Uganda	UG	
Ukraine	UA	
United Arab Emirates	AE	
United Kingdom	GB	
United Republic of Tanzania	TZ	
United States	US	
Uruguay	UY	
Uzbekistan	UZ	
Vanuatu	VU	
Viet Nam	VN	
Wallis and Futuna	WF	
Yemen	YE	
Zambia	ZM	
Zimbabwe	ZW	

Related Information

[Global Address: Suggestion List Options \[page 583\]](#)

8.5.13.5 Information codes (Global Address Cleanse)

Information codes are four characters that explain why an address is unassigned. Information codes have six levels of classification:

- The 1000 level represents input record discrepancies.

- The 2000 level represents inconsistent last line information.
- The 3000 level represents inconsistent address information.
- The 4000 level represents inconsistent secondary address information.
- The 5000 level represents all other types of information.
- The 6000 level represents an unclassified error.
- The 7000 level represents an aborted process due to a user-defined timeout.

The table also shows that each information code is available based on the engine(s) that you enable.

- Canada (C)
- Global Address (G)
- USA (U)
- All engines: Consists of C, G, and U.
- Transform Level (T): Information code does not come from a specific engine.

Use the following table to determine the code assigned to the Info_Code output field.

Information code	Description	Engine(s)
1020	Address validated in multiple countries.	T
1030	No country found by Country ID or no country set for the record.	T
1040	Address contains at least one character that is not part of the character set supported by the engine.	T
1060	The country identified is not supported by any of the active engines.	T
1080	The script identified is not supported by any of the active engines.	T
2000	Unable to identify locality, region, and/or postcode information on input.	All engines
2010	Unable to identify locality and invalid postcode found.	All engines
2020	Unable to identify postcode. Invalid locality is preventing a possible address correction.	All engines
2030	Invalid locality and postcode are preventing a possible address correction.	All engines
2040	Invalid postcode is preventing a locality selection.	G, U
2050	Lastline matches are too close to choose one.	G
2070	Bad or no postcode for locality; multiple postcodes are available.	G
3000	Locality, region, and postcode are valid. Unable to identify the primary address line.	All engines
3010	Locality, region, and postcode are valid. Unable to match primary name to directory.	All engines
3020	Possible primary name matches are too close to choose one.	All engines

Information code	Description	Engine(s)
3030	Primary range is missing on input or not in the directory.	All engines
3050	An invalid or missing primary type is preventing a possible address match.	All engines
3060	A missing primary type and prefix/postfix (directional) is preventing a possible address match.	G, U
3070	An invalid or missing prefix/postfix (directional) is preventing a possible address match.	All engines
3080	An invalid or missing postcode is preventing a possible address match.	All engines
3090	An invalid or missing locality is preventing a possible address match.	G, U
3100	Possible address-line matches are too close to choose one.	All engines
3110	Address conflicts with postcode and the same primary name has a different postcode.	C
3200	The building name is missing on input or not in the directory.	G
3210	The building's address is not in the directory.	G
3220	Possible building names are too close to choose one.	G
3250	The building name is missing on input or not in the directory, or the range is missing on input or invalid with the input building.	G
3300	The postcode only lookup returned multiple primary names.	G
4000	The secondary information is missing on input or not in the directory.	All engines
4010	Possible secondary address line matches are too close to choose one.	All engines
4500	The organization is missing on input or not in the directory.	G
4510	The organization's address is not in the directory.	G
4520	Possible organization names are too close to choose one.	G
5000	The address was valid, but the postal authority classified this address as undeliverable.	G, U
5010	The address does not reside in the specified country.	C, U
5020	The entire input address was blank.	T
5030	The country's postal authority will not permit assignment due to violation of an assignment rule.	G

Information code	Description	Engine(s)
5040	Assignment not made after lastline processing due to a violation of lastline assignment rules.	G
5050	The obsolete address can be mapped to multiple addresses.	G
6000	Unclassified error.	All engines
7000	The user-defined processing timeout threshold (set in the Processing Timeout option) has been reached, preventing further corrective action on this address.	G

8.5.13.6 Status codes (Global Address Cleanse)

Status codes (assigned to the Status_Code output field) are five or six characters that represent the corrections made to the address during processing. The number of characters depends on the engine used for processing.

- The first character is always S (for Status).
- The second character is associated with any last line corrections.
- The third character is associated with any address line corrections.
- The fourth character is associated with any secondary address line corrections.
- The fifth character is associated with changes to components that are not considered basic address components (Other Primary Address and Other Secondary Address).
- The sixth component indicates additional information about a record that is not related to a change in the address.

First character

The value of the first character is always S for Status.

Second character

The value of the second character depends on corrections to the country, postcode, region, or locality.

Value	Description
0	No significant difference between the input data and the corrected data.
1	Corrected country.
2	Corrected postal code.

Value	Description
3	Corrected country and postal code.
4	Corrected region.
5	Corrected country and region.
6	Corrected postal code and region.
7	Corrected country, postal code, and region.
8	Corrected locality.
9	Corrected country and locality.
A	Corrected postal code and locality.
B	Corrected country, postal code, and locality.
C	Corrected region and locality.
D	Corrected country, region, and locality.
E	Corrected postal code, region, and locality.
F	Corrected country, postal code, region, and locality.

Third character

The value of the third character depends on corrections to the pre/post directionals, primary type, primary name, and primary range.

Value	Description
0	No significant difference between the input data and the corrected data.
1	Corrected pre/post directional.
2	Corrected primary type.
3	Corrected pre/post directional and primary type.
4	Corrected primary name.
5	Corrected pre/post directional and primary name.
6	Corrected primary type and primary name.
7	Corrected pre/post directional, primary type, and primary name.
8	Corrected primary range.
9	Corrected pre/post directional and primary range.
A	Corrected primary type and primary range.
B	Corrected pre/post directional, primary type, and primary range.
C	Corrected primary name and primary range.
D	Corrected pre/post directional, primary name, and primary range.

Value	Description
E	Corrected primary type, primary name, and primary range.
F	Corrected pre/post directional, primary type, primary name, and primary range.

Fourth character

The value of the fourth character depends on corrections to the unit description, unit number, firm, building name, floor description, floor number, stairwell description, stairwell name, wing description, and wing name.

Value	Description
0	No significant difference between the input data and the corrected data.
1	Corrected one or more secondary address component (unit description, floor description, stairwell description, or wing description).
2	Corrected one or more secondary address component (unit number, floor number, stairwell name, or wing name).
3	Corrected one or more secondary address component (unit description, unit number, floor description, floor number, stairwell description, stairwell name, wing description, or wing name).
4	Corrected building name.
5	Corrected one or more secondary address component (unit description, floor description, stairwell description, or wing description), and building name.
6	Corrected the one or more secondary address component (unit number, floor number, stairwell name, or wing name), and building name.
7	Corrected one or more secondary address component (unit description, unit number, floor description, floor number, stairwell description, stairwell name, wing description, or wing name), and building name.
8	Corrected firm.
9	Corrected one or more secondary address component (unit description, floor description, stairwell description, or wing description), and firm.
A	Corrected one or more secondary address component (unit number, floor number, stairwell name, or wing name), and firm.
B	Corrected one or more secondary address component (unit description, unit number, floor description, floor number, stairwell description, stairwell name, wing description, or wing name), and firm.
C	Corrected building name and firm.
D	Corrected one or more secondary address component (unit description, floor description, stairwell description, or wing description), building name, and firm.
E	Corrected one or more secondary address component (unit number, floor number, stairwell name, or wing name), building name, and firm.
F	Corrected one or more secondary address component (unit description, unit number, floor description, floor number, stairwell description, stairwell name, wing description, or wing name), building name, and firm.

Fifth character

The value of the fifth character depends on changes to components that are not considered basic address components (Other Primary Address and Other Secondary Address).

Other Primary Address components:

- Primary_Delivery_Mode
- Primary_Delivery_Number

Other Secondary Address components:

- Delivery_Installation_Name
- Delivery_Installation_Qualifier
- Delivery_Installation_Type

Value	Description
0	No significant change between the input data and the corrected data.
1	Changed the Other Primary Address components.
2	Changed the Other Secondary Address components.
3	Changed the Other Primary Address and Other Secondary Address components.
4	Changed the point of reference.
5	Changed the Other Primary Address components and the point of reference.
6	Changed the Other Secondary Address components and the point of reference.
7	Changed the Other Primary Address, Other Secondary Address components, and the point of reference.

Sixth character

The value of the sixth character indicates additional information about a record that is not related to a change in the address.

Value	Description
A	Alias record used for assignment. Global Address engine.
B	Base record assignment. Global Address engine (New Zealand). A Bordering Locality. Global Address engine (Australia).
C	An Alias and a Bordering locality. Global Address engine (Australia).
D	Deleted record. Global Address engine (Austria and Germany).
E	An excluded postal code was found. Canada engine.
I	Record ignored. Global Address engine (New Zealand).
L	Large Volume Receiver (LVR). Global Address engine (Brazil).

Value	Description
U	Unique address. Global Address engine (New Zealand).

8.5.13.7 Quality codes (Global Address Cleanse)

Quality codes relay additional information about the quality of the address. There are six levels of quality codes based on these factors:

- The country of the input data
- The engine used for processing
- The information code
- The status code if there is not an information code

Use the following table to determine the code assigned to the Quality_Code output field.

Quality code	Description
Q1	Perfect address on input. All address components were validated without corrections.
Q2	Corrected address. All address components were validated after corrections were made.
Q3	Not all components of the address could be fully validated. There was insufficient information to make a final correction. However, the assessment of the record leads to the assumption that there is a "high" likelihood that this address is deliverable.
Q4	Not all components of the address could be fully validated. There was insufficient information to make a final correction. However, the assessment of the record leads to the assumption that there is a "fair" likelihood that this address is deliverable.
Q5	Not all components of the address could be fully validated. There was insufficient information to make a final correction. However, the assessment of the record leads to the assumption that there is a "small" likelihood that this address is deliverable.
Q6	Not all components of the address could be fully validated. There was insufficient information to make a final correction. However, the assessment of the record leads to the assumption that it is "highly unlikely" that this address is deliverable.

8.5.13.8 Fault codes (USA Regulatory Address Cleanse)

When the transform cannot assign an address, it creates a fault code (Fault_Code output field). This code tells you why the address could not be assigned.

Fault Code	Description
E101	Last line is bad or missing.
E212	No locality and bad postal code.
E213	Bad locality, valid region, and no postal code.

Fault Code	Description
E214	Bad locality and bad postal code.
E216	Bad postal code and can't determine which locality match to select.
E302	No primary address line parsed.
E412	Primary name not found in directory.
E413	Possible primary name matches are too close to choose one.
E420	Primary range is missing.
E421	Primary range is invalid for the street/route/building.
E422	Primary prefix needed, input is wrong or missing.
E423	Primary type needed, input is wrong or missing.
E425	Primary type and directional needed, input is wrong or missing.
E427	Primary postfix needed, input is wrong or missing.
E428	Bad postal code, can't select an address match.
E429	Bad locality, can't select an address match.
E430	Possible address-line matches too close to choose one.
E431	Locality2 needed, input is wrong or missing.
E439	Exact match made in EWS directory.
E500	Other error.
E501	Foreign address.
E502	Input record entirely blank.
E503	Postal code not in area covered by partial USPS directory.
E504	Overlapping ranges in USPS directory.
E505	Address does not exist in the USPS directories. Undeliverable address.
E600	Marked by USPS as unsuitable for delivery of mail.
E601	The primary address number did not DPV confirm, and the Postcode2 was removed.

8.5.13.9 Status codes (USA Regulatory Address Cleanse)

When the transform assigns an address, it creates a status code (Status_Code output field). This code can tell you how the input address differs from the assigned address.

Digit	Description
1st	<p><i>A</i>: The transform truncated the address line to make it fit your field.</p> <p><i>B</i>: The transform truncated both the address line and the Locality1_Name.</p> <p><i>C</i>: The transform truncated the Locality1_Name to make it fit your field.</p> <p><i>S</i>: No truncation occurred.</p>
2nd	<p><i>0</i>: Regarding the Locality1_Name, Region1, Postcode1, and Postcode2, there is no significant difference between the input data and the data that the transform assigned.</p> <p><i>1</i>: The transform assigned a different Postcode1.</p> <p><i>2</i>: The transform assigned a different Locality1_Name.</p> <p><i>3</i>: The transform assigned a different Locality1_Name and Postcode1.</p> <p><i>4</i>: The transform assigned a different Region1.</p> <p><i>5</i>: The transform assigned a different Region1 and Postcode1.</p> <p><i>6</i>: The transform assigned a different Locality1_Name and Region1.</p> <p><i>7</i>: The transform assigned a different Locality1_Name, Region1, and Postcode1.</p> <p><i>8</i>: The transform assigned a different Postcode2.</p> <p><i>9</i>: The transform assigned a different Postcode1 and Postcode2.</p> <p><i>A</i>: The transform assigned a different Locality1_Name and Postcode2.</p> <p><i>B</i>: The transform assigned a different Locality1_Name, Postcode1, and Postcode2.</p> <p><i>C</i>: The transform assigned a different Region1 and Postcode2.</p> <p><i>D</i>: The transform assigned a different Region1, Postcode1, and Postcode2.</p> <p><i>E</i>: The transform assigned a different Locality1_Name, Region1, and Postcode2.</p> <p><i>F</i>: The transform assigned a different Locality1_Name, Region1, Postcode1, and Postcode2.</p>

Digit	Description
3rd	<p><i>0</i>: Regarding the primary name, primary prefix/postfix, and primary type, there is no significant difference between the input and what the transform assigned.</p> <p><i>1</i>: The transform assigned a different primary type.</p> <p><i>2</i>: The transform assigned a different primary prefix.</p> <p><i>3</i>: The transform assigned a different primary prefix and primary type.</p> <p><i>4</i>: The transform assigned a different primary postfix.</p> <p><i>5</i>: The transform assigned a different primary type and primary postfix.</p> <p><i>6</i>: The transform assigned a different primary prefix and primary postfix.</p> <p><i>7</i>: The transform assigned a different primary prefix, primary type, and primary postfix.</p> <p><i>8</i>: The transform assigned a different primary name.</p> <p><i>9</i>: The transform assigned a different primary name and primary type.</p> <p><i>A</i>: The transform assigned a different primary prefix and primary name.</p> <p><i>B</i>: The transform assigned a different primary prefix, primary name, and primary type.</p> <p><i>C</i>: The transform assigned a different primary name and primary postfix.</p> <p><i>D</i>: The transform assigned a different primary name, primary type, and primary postfix.</p> <p><i>E</i>: The transform assigned a different primary prefix, primary name, and primary postfix.</p> <p><i>F</i>: The transform assigned a different primary prefix, primary name, primary postfix, and primary type.</p>

Digit	Description
4th	<p><i>0</i>: Regarding the county number, sort code route, delivery point, and unit description, there is no significant difference between the input data and the data that the transform assigned.</p> <p><i>1</i>: The transform assigned a different unit description.</p> <p><i>2</i>: The transform assigned a different delivery point.</p> <p><i>3</i>: The transform assigned a different delivery point and unit description.</p> <p><i>4</i>: The transform assigned a different sort code route.</p> <p><i>5</i>: The transform assigned a different sort code route and unit description.</p> <p><i>6</i>: The transform assigned a different sort code route and delivery point.</p> <p><i>7</i>: The transform assigned a different sort code route, delivery point, and unit description.</p> <p><i>8</i>: The transform assigned a different county number.</p> <p><i>9</i>: The transform assigned a different county number and unit description.</p> <p><i>A</i>: The transform assigned a different county number and delivery point.</p> <p><i>B</i>: The transform assigned a different county number, delivery point, and unit description.</p> <p><i>C</i>: The transform assigned a different county number and sort code route.</p> <p><i>D</i>: The transform assigned a different county number, sort code route, and unit description.</p> <p><i>E</i>: The transform assigned a different county number, sort code route, and delivery point.</p> <p><i>F</i>: The transform assigned a different county number, sort code route, delivery point, and unit description.</p>
5th	<p><i>0</i>: Regarding the LOT, LOT_Order, and Locality2_Official, there is no significant difference between the input data and the data that the transform assigned.</p> <p><i>1</i>: The transform assigned a different LOT.</p> <p><i>2</i>: The transform assigned a different LOT_Order.</p> <p><i>3</i>: The transform assigned a different LOT and LOT_Order.</p> <p><i>4</i>: The transform assigned a different Locality2_Official.</p> <p><i>5</i>: The transform assigned a different Locality2_Official and LOT.</p> <p><i>6</i>: The transform assigned a different Locality2_Official and LOT_Order.</p> <p><i>7</i>: The transform assigned a different Locality2_Official, LOT, and LOT_Order.</p>
6th	Always outputs a zero (<i>0</i>).

8.5.13.10 About ShowA and ShowL (USA)

The Show programs are used for looking inside the postal directories to find answers to questions like these:

- Why did the transform standardized the address in an unexpected way?
- Why didn't the transform assign the address?
- Why did the transform's error code indicate a flaw in the directory?

i Note

Run the ShowA/ShowL utilities from a DOS command line using specific command-line options. These options are listed when you enter the following command:

Windows: `showa /op`

UNIX: `showa -op`

Query the postal directories

You can use ShowA to display or output information from the Address_1_Directory, and you can use ShowL to query the City_Directory and the Post_Code_Directory.

Edit configuration files

Each Show utility has its own configuration file. These files contain parameters for controlling how the program behaves.

USA addresses:

Utility	Executable	File name	Location
ShowA	showa.exe	showa.cfg	<LINK_DIR>\dataquality\urac
ShowL	showl.exe	showl.cfg	<LINK_DIR>\dataquality\urac

Before you run the Show utilities, set both configuration files for the appropriate country directory. The configuration files contain instructions and detailed information about how to run the programs.

i Note

Run the ShowA/ShowL utilities in the same directory as the ShowA/ShowL configuration files. You can change the location of the Show A/L executable files, however the utilities will not run if you did not accept the default location for the configuration files.

Related Information

[USA ShowA command line options \[page 821\]](#)

8.5.13.10.1 ShowA/ShowL modes of operation

The ShowA and ShowL utilities have two modes for entering queries: prompts and command line options.

Prompts

If you type the ShowA/ShowL command without any options at all, then ShowA/ShowL prompts you to enter your query data. It takes all other information and options from the configuration file.

After your first query, ShowA/ShowL prompts you to enter the next one. You can exit by typing "quit" at any prompt.

Command line options

You may enter your query data on your ShowA/ShowL command line. ShowA/ShowL performs one query, displays and/or outputs the results, and then exits.

From the configuration file, ShowA/ShowL takes information about the auxiliary files, output file, display and search options. If you must override any value taken from the configuration file, you can do so. You can use command line options selectively to override where you need to, and depend on the configuration file for the rest. The only value you cannot specify through command- line options is the optional output fields.

i Note

If any command option is present, then ShowA/ShowL detects that you are operating in command line mode. It will not prompt you to enter your query data. Use command line options to enter your query data.

8.5.13.10.2 USA ShowA command line options

To view a summary of command line options, use this command:

Windows: `showa /op`

UNIX: `showa -op`

The following table lists the command line options and the command descriptions.

UNIX	Windows	Description
-a	/a	Appends information to the output file (if it already exists).
-alias	/alias	Includes preferred alias address lines.
-d	/d	Displays your query data on screen.
-fin	/fin	Expands the query to cover USPS finance area.

UNIX	Windows	Description
-op	/op	Displays the list of options (in this table).
-p	/p	Pauses screen display every 22 lines.
-2:dpbc	/2:dpbc	Enter the DPBC code for <code>dpbc</code> .
-4:zip4	/4:zip4	Enter the postcode2 for <code>zip4</code> .
-ad:file	/ad:file	Enter the Address-line dictionary and path name (addrln.dct) for <code>file</code> .
-c:cart	/c:cart	Enter the carrier route number for <code>cart</code> .
-f:file	/f:file	Enter the file path and name of the output file (to hold the information from the query instead of just displaying it on screen) for <code>file</code> .
-nd:file	/nd:file	Enter the National ZIP+4 directory path and name (zip4us.dir) for <code>file</code> .
-pre:dir	/pre:dir	Enter the primary prefix (N, NE, E, SE, S, SW, W, NW) for <code>dir</code> .
-pos:dir	/pos:dir	Enter the primary postfix (N, NE, E, SE, S, SW, W, NW) for <code>dir</code> .
-s:street	/s:street	Enter the street primary name (in quotes if multiple words) for <code>street</code> .
-sfx:suffix	/sfx:suffix	Enter the primary type (Ave, Blvd, St, Rd, and so on) for <code>suffix</code> .
-sh:range	/sh:range	Enter the street (primary) range high for <code>range</code> .
-sl:range	/sl:range	Enter the street (primary) range low or exact for <code>range</code> .
-t:type	/t:type	Enter the file type (dBASE3, ASCII, or DELIMITED) for <code>type</code> .
-u:urb idx	/u:urb idx	Enter the urbanization Index for <code>urb idx</code> .
-z:lo-hi	/z:lo-hi	Enter the low and high range for postcode1 for <code>lo-hi</code> .
-z:zip	/z:zip	Enter the postcode1 for <code>zip</code> .

Related Information

[ShowA/ShowL modes of operation \[page 821\]](#)

8.5.13.10.3 USA ShowL command line options

To view a summary of command line options, use this command:

Windows: `showl /op`

UNIX: `showl -op`

The following table lists the command line option and the command descriptions.

UNIX	Windows	Description
-a	/a	Appends query information to an output file (if it already exists).
-d	/d	Displays query data on screen.
-op	/op	Displays this list of options.
-p	/p	Pauses screen display every 22 lines.
-ab:query	/ab:query	Enter the abbreviated locality1 for query.
-cd:file	/cd:file	Enter the City directory path and name (city04.dir) for file.
-cn:city	/cn:city	Enter locality1 name (in quotes if multiple words) for city.
-dr:dir	/dr:dir	Enter which directory to search, City or ZCF for dir.
-f:file	/f:file	Enter the output file path and name for file.
-ml:query	/ml:query	Enter the military postcode1 for query.
-mz:query	/mz:query	Enter the multi-zone locality1 for query.
-pn:query	/pn:query	Enter a place name for query.
-st:state	/st:state	Enter region1 (for US use USPS abbreviations or full state names) for state.
-t:type	/t:type	Enter the output file type (dBASE3, ASCII, or DELIMITED) for type.
-un:query	/un:query	Enter the unique postcode1 for query.
-z:zip	/z:zip	Enter the postcode1 for zip.
-zd:file	/zd:file	Enter the ZCF directory path and name (zcf04.dir) for file.
-z:lo-hi	/z:lo-hi	Enter the postcode1 range for lo-hi.

Related Information

[ShowA/ShowL modes of operation \[page 821\]](#)

8.5.14 Data Cleanse reference

This section describes the following reference information for use with the Data Cleanse transform:

- Status codes
- Information codes
- Diacritical character conversion chart

Parent topic: [Data Quality transforms \[page 437\]](#)

Related Information

[Blueprints and other content objects for download \[page 438\]](#)

[About Data Quality fields \[page 445\]](#)

[About data quality statistics \[page 448\]](#)

[Associate \[page 454\]](#)

[Country ID \[page 472\]](#)

[Data Cleanse \[page 474\]](#)

[DSF2® Walk Sequencer \[page 514\]](#)

[Geocoder \[page 524\]](#)

[Global Address Cleanse \[page 552\]](#)

[Global Suggestion List \[page 640\]](#)

[Match \[page 651\]](#)

[USA Regulatory Address Cleanse \[page 710\]](#)

[Address Cleanse reference \[page 765\]](#)

8.5.14.1 Status codes (Data Cleanse)

Status codes (assigned to the Status_Code output field) have the same format: `<parser name>_<output field name>_STD` or `<parser name>_<output field name>_ADD`. The `<parser name>` includes one of the following parsers: custom, date, firm, person, phone, and SSN.

Only the status codes that have standardized output are shown. The status codes show one of the following depending on whether the standardization options are chosen:

- how the data was standardized
- that a specific standard could be used on the data

i Note

When only minor changes occur, a status code may not be generated. See the exceptions at the bottom of this topic.

Person parser

Person parser status codes have the `PERSON#_<output field>_STD` or `PERSON#_<output field>_ADD` format where the pound sign (#) is replaced with the person number, such as Person1 or Person2 through Person6.

Person parser status codes

Status code	Description
Person#_Name_Designator_Std	Used a Name_Designator standard.

Status code	Description
Person#_Prename_Add	Prename was assigned.
Person#_Prename_Std	Prename was provided and a prename standard was used.
Person#_Given_Name1_Std	Used a Given_Name1 standard.
Person#_Given_Name2_Std	Used a Given_Name2 standard.
Person#_Family_Name1_Std	Used a Family_Name1 standard.
Person#_Family_Name2_Std	Used a Family_Name2 standard.
Person#_Honorary_Postname_Std	Used an Honorary_Postname standard.
Person#_Maturity_Postname_Std	Used a Maturity_Postname standard.
Person#_Title_Std	Used a Title standard.
Person#_Name_Special_Std	Used a Name_Special standard.

❖ Example

Multiple status codes can be assigned per record and are separated by a comma. For example, the status code:

```
PERSON1_PRENAME_STD, PERSON1_GIVEN_NAME1_STD, PERSON1_FAMILY_NAME1_STD, PERSON2_GIVEN_NAME2_STD, PERSON6_MATURITY_POSTNAME_STD
```

provides the following information:

- The Person1 parse used standards for Prename, Given_Name1, and Family_Name1 output fields.
- The Person2 parse used standards for the Given_Name2 output field.
- The Person6 parse used standards for the Maturity_Postname output field.

Firm parser

Firm parser status codes have the FIRM#_<output field>_STD format where the pound sign (#) is replaced with the Firm number, such as Firm1 or Firm2 through Firm6.

Firm parser status codes

Status code	Description
Firm#_Firm_Std	Used a Firm standard.
Firm#_Firm_Location_Std	Used a Firm_Location standard.

❖ Example

Multiple status codes can be assigned per record and are separated by a comma. For example, the status code:

```
FIRM1_FIRM_STD, FIRM2_FIRM_LOCATION_STD, FIRM3_FIRM_STD, FIRM3_FIRM_LOCATION_STD
```

provides the following information:

- The Firm1 parse used a Firm standard.
- The Firm2 parse used a Firm_Location standard.
- The Firm3 parse used both a Firm standard and a Firm_Location standard.

Date, Phone and Social Security number parsers

Date, Phone, and Social Security number (SSN) parser status codes have the following status code formats:

Date, Phone and Social Security number parser status codes

Status code	Description
Date#_Date_Std	Used a Date standard.
Phone#_Phone_Std	Used a Phone number standard.
SSN#_SSN_Std	Used a Social Security number standard.

❖ Example

Multiple status codes can be assigned per record and are separated by a comma. For example, the Date status code, DATE1_DATE_STD, DATE6_DATE_STD, shows that both Date1 and Date6 were standardized.

SSN1_SSN_STD, SSN5_SSN_STD shows that both SSN1 and SSN5 were standardized.

PHONE1_PHONE_STD, PHONE3_PHONE_STD shows that both Phone1 and Phone3 were standardized.

Custom parser

Custom parser status codes have the `<parser name>#_<output field>_STD` format where the pound sign (#) is replaced with the parser number.

Custom parser status code

Status code	Description
<code><parser name>#_<output field>_Std</code>	Used the <code><output field></code> 's standard.

❖ Example

Multiple status codes can be assigned per record and are separated by a comma. For example, the Automobile status code, AUTOMOBILE1_COLOR_STD, AUTOMOBILE1_SIZE_STD, AUTOMOBILE2_COLOR_STD, shows the following information:

- Automobile1 used a color and size standard.
- Automobile2 used a color standard.

Status code exceptions

When there are only minor changes, a status code may not be generated. There are several instances where a status code is not generated.

Exception	Description	Example
Trailing period	A status code is not generated when a period is added or removed after a piece of data.	J Smith has a standard of "J." No status code. J. Smith has a standard of "J" No status code. UK has a standard of "U.K." A status code is generated because there are more punctuation changes than the trailing period.
<i>Capitalization</i> option	A status code is not generated when the only difference between the standard and the original data is how the data is cased.	Mcdonald has a standard of "McDonald" but is output as "MCDONALD" because the <i>Capitalization</i> option is set to <i>Upper</i> . No status code.
<i>Remove punctuation</i> option	A status code is not generated when the only difference between the standard and the original data is removed punctuation when this option is set to <i>Yes</i> .	UK has a standard of "U.K." However, the <i>Remove punctuation</i> option is set to <i>Yes</i> . No status code.
<i>Remove diacritical characters</i> option	A status code is not generated when the only difference between the standard and the original data is the diacritical characters when this option is set to <i>Yes</i> .	Hernandez has a standard of "Hernández". However, the <i>Remove diacritical characters</i> option is set to <i>Yes</i> . No status code. Hernandez has a standard of "Hernández". However, the <i>Remove diacritical characters</i> option is set to <i>No</i> . A status code is generated to show that a standard was used.

8.5.14.2 Information codes (Data Cleanse)

Information codes (assigned to the Info_Code output field) provide information about the data that may be suspect and might require a manual review. The output field contains one or more codes separated by a comma. Each code begins with a letter followed by three numbers classified as follows:

Date parse information

Information code format	Description
D1##	Date 1 parse level information
D2##	Date 2 parse level information
D3##	Date 3 parse level information
D4##	Date 4 parse level information
D5##	Date 5 parse level information
D6##	Date 6 parse level information

Firm parse information

Information code format	Description
F1##	Firm 1 parse level information
F2##	Firm 2 parse level information
F3##	Firm 3 parse level information
F4##	Firm 4 parse level information
F5##	Firm 5 parse level information
F6##	Firm 6 parse level information

Input field information

Information code format	Description
I###	Input field level information

Person parse information

Information code format	Description
P1##	Person 1 parse level information
P2##	Person 2 parse level information
P3##	Person 3 parse level information
P4##	Person 4 parse level information
P5##	Person 5 parse level information
P6##	Person 6 parse level information

Phone parse information

Information code format	Description
T1##	Phone 1 parse level information
T2##	Phone 2 parse level information
T3##	Phone 3 parse level information
T4##	Phone 4 parse level information
T5##	Phone 5 parse level information
T6##	Phone 6 parse level information

Record level information

Information code format	Description
R###	Record level information

Related Information

[Detailed descriptions of information codes \[page 829\]](#)

8.5.14.2.1 Detailed descriptions of information codes

The following table shows more details about Data Cleanse information codes.

Information code	Description
R001	All input field data went to one or more Extra output fields; nothing was parsed for the record.
R002	Parsed some of the input fields. One or more other input fields went to the Extra output field.
R003	Parsed part of the input fields. Some of the record data went to the Extra output field.
R004	All of the record's input fields were parsed into appropriate output fields. None of the input data went to the Extra output field.
R400	The data in Option_Content_Domain_Sequence overrode the content domain sequence transform option.
R405	The data in Option_Content_Domain_Sequence was not recognized as a content domain sequence. The data in Option_Country or the data in content domain sequence transform option was used to determine the content domain sequence.
R410	The data in Option_Output_Format overrode the output format data cleanse transform option.
R415	The data in Option_Output_Format was not recognized as a valid output format. The data in Option_Country or the output format transform option was used to determine the output format.
R420	The data in Option_Country was recognized as an ISO2 country code and overrode the content domain sequence transform option.
R421	The data in Option_Country was recognized as an ISO2 country code and overrode the output format transform option.
R425	<p>The data in Option_Country was not recognized as an ISO2 country code. One or more situations occurred where Option_Country data was used to override the content domain sequence and/or output format transform options, but failed:</p> <ul style="list-style-type: none">• Attempted to use Option_Country to override the content domain sequence transform option. This occurs when the Option_Content_Domain_Sequence data is invalid (R405) or the data is not supplied (no status code is generated when this occurs). The data was parsed based using the content domain sequence transform option.• Attempted to use Option_Country to override the output transform option. This occurs when the Option_Output_Format data is invalid (R415) or is not supplied (no status code is generated when this occurs). The data was parsed based using the output format transform option.
R428	The cleansing package does not recognize the country code provided in the Option_Country input field.
P#01	The person# parse contained some data that was not found in the cleansing package. This information code does not report on title information, which is different than information code P#51.
P#02	The person# parse had a close firm parse. This is only applicable for Person_Firm multiline parse when using the Person_Firm multiline parser or Name_Firm_Line when the data came from the input field Name_Firm_Line.
P#03	The person# parse was a presumptive name parse (based on reasonable evidence).
P#04	The person# parse has no given name, or has a questionable given name.
P#05	The person# parse has no family name, or has a questionable family name.
P#51	The person# parse contained a title token that was not found in the cleansing package. This is different than information code P#01.

Information code	Description
P#52	The title# parse has a close firm parse.
P#53	The title# parse was presumptive.
F#01	The firm# parse contained some data that was not found in the cleansing package.
F#02	The firm# parse had a close person parse. This is only applicable for Person_Firm multiline parse when using the Person_Firm multiline parser or Name_Firm_Line when the data came from the input field Name_Firm_Line.
F#03	The firm# parse was a presumptive firm parse (based on reasonable evidence).
D#01	Date# was not in the expected format.
D#02	Date# was converted from 2-digits to 4-digits. The century threshold transform option was applied.
D#03	Date# was in the Day_Month_Year format.
D#04	Date# was in the Month_Day_Year format.
D#05	Date# was in the Year_Month_Day format.
D#06	Date# was in the Year_Day_Month format.
D#07	Date# was in an ambiguous format. There is more than one possible format for the date. For example, 12/09/10 is valid for all of the formats, whereas 03/16/94 is valid only for the Month_Day_Year format.
T#01	The phone# parse did not have a North American area code.
T#02	The transform parsed phone data using a different country than the country listed in the Option_Country field.
T#03	The transform parsed phone data by prepending a country code to the incoming phone data.
I111-I116	All of the input data in Name_Line# went to one or more Extra output fields; nothing was parsed for this input field.
I121-I126	There was no input data in Name_Line#.
I131-I136	Parsed some of the input data in Name_Line#, the rest of the data is in the Extra output field.
I151-I156	All of the input data in Title_Line# went to one or more Extra output fields. Nothing was parsed for this input field.
I161-I166	There was no input data in Title_Line#.
I171-I176	Parsed some of the input data in Title_Line#, the rest of the data is in the Extra output field.
I211-I212	All of the input data in Person# discrete went to one or more Extra output fields. Nothing was parsed for this input field.
I221-I222	There was no input data in Person# discrete.
I231-I232	Parsed some input data from Person# discrete. Remaining data is in the Extra output field.
I311-I316	All of the input data in Name_Or_Firm_Line# went to one or more Extra output fields; nothing was parsed for this input field.
I321-I326	There was no input data in Name_Or_Firm_Line#.

Information code	Description
I331-I336	Parsed some of the input data in Name_Or_Firm_Line#, the rest of the data is in the Extra output field.
I351-I352	All of the input data in Firm_Line# went to one or more Extra output fields; nothing was parsed for this input field.
I361-I366	There was no input data in Firm_Line#.
I371-I372	Parsed some of the input data in Firm_Line#, the rest of the data is in the Extra output field.
I421-I422	There was no input data in Firm# discrete.
I511-I516	All of the input data in Date# went to one or more Extra output fields; nothing was parsed for this input field.
I521-I526	There was no input data in Date#.
I531-I536	Parsed some of the input data in Date#, the rest of the data is in the Extra output field.
I711-I716	All of the input data in Email# went to one or more Extra output fields; nothing was parsed for this input field.
I721-I726	There was no input data in Email#.
I731-I736	Parsed some of the input data in Email#, the rest of the data is in the Extra output field.
I751-I756	All of the input data in Phone# went to one or more Extra output fields; nothing was parsed for this input field.
I761-I766	There was no input data in Phone#.
I771-I776	Parsed some of the input data in Phone#, the rest of the data is in the Extra output field.
I811-I816	All of the input data in SSN# went to one or more Extra output fields; nothing was parsed for this input field.
I821-I826	There was no input data in SSN#.
I831-I836	Parsed some of the input data in SSN#, the rest of the data is in the Extra output field.
I851-I856	All of the input data in UDPM# went to one or more Extra output fields; nothing was parsed for this input field.
I861-I866	There was no input data in UDPM#.
I871-I876	Parsed some of the input data in UDPM#, the rest of the data is in the Extra output field.
I901-I912	All of the input data in Date# went to one or more Extra output fields; nothing was parsed for this input field.
I931-I942	There was no input data in Multiline#.
I951-I962	Parsed some of the input data in Multiline#, the rest of the data is in the Extra output field.

8.5.14.3 Diacritical character conversion chart

In the Data Cleanse transform, there is an option where you can remove the diacritical characters from the data on output. The following table shows the diacritical characters and the ASCII equivalent letter that replaces the diacritical character.

Note

The diacritical character is listed first, separated by a comma and then the ASCII replacement letter.

Diacritical characters conversion chart

À,A	Ě,E	Ł,L	Ś,S	Ž,Z	ē,e	İ,I	Š,s
Á,A	Ě,E	Ł,L	Ŝ,S	Ž,Z	ě,e	ı,ı	š,s
Â,A	Ě,E	Ł,L	Ș,S	à,a	é,e	ı,ı	ț,t
Ã,A	Ĝ,G	Ł,L	Ț,T	á,a	ę,e	ñ,n	ț,t
Ä,A	Ğ,G	Ń,N	Ț,T	â,a	ğ,g	ń,n	ț,t
Å,A	Ğ,G	Ń,N	Ț,T	ã,a	ğ,g	ņ,n	ù,u
Ă,A	Ğ,G	Ń,N	Ț,T	ä,a	ğ,g	ň,n	ú,u
Ä,A	Ĥ,H	Ń,N	Ú,U	å,a	ğ,g	ò,o	û,u
Ȧ,A	Ĥ,H	Ò,O	Û,U	ä,a	ĥ,h	ó,o	ü,ue
Ç,C	İ,I	Ó,O	Ü,UE	ǎ,a	ħ,h	ô,o	ů,u
Ć,C	Í,I	Ô,O	Ů,U	ą,a	ı,i	õ,o	ů,u
Ĉ,C	Î,I	Õ,O	Ů,U	ç,c	í,i	ö,oe	ů,u
Č,C	Ĭ,I	Ö,OE	Ů,U	ć,c	î,i	ø,o	ů,u
Č,C	Ĭ,I	Ø,O	Ů,U	ĉ,c	ı,i	ö,o	ů,u
Đ,D	Ĭ,I	Ō,O	Ů,U	ć,c	ı,i	ö,o	ŵ,w
Ď,D	Ĭ,I	Ō,O	Ů,U	č,c	ı,i	ö,o	ý,y
È,E	Ĭ,I	Ō,O	Ŵ,W	ď,d	ı,i	ř,r	ý,y
É,E	Ĭ,I	Ř,R	Ý,Y	è,e	ı,i	ř,r	ý,y
Ê,E	Ĭ,I	Ř,R	Ÿ,Y	é,e	ı,i	ř,r	ž,z
Ë,E	Ĭ,I	Ř,R	Ÿ,Y	ê,e	ı,i	š,s	ž,z
Ê,E	Ĭ,I	Š,S	Ž,Z	ë,e	ı,i	š,s	ž,z

Example

This example shows the input data and how the diacritical characters are replaced on output.

Diacritical character removal example

Input	Output
María Hernández	Maria Hernandez
Geschäftsführer	Geschaeftsfuehrer

Input	Output
Telecomunicações São Paulo	Telecomunicacoes Sao Paulo

Related Information

[Other standardization options \[page 493\]](#)

8.6 Platform transforms

Transforms that are needed for general data movement operations.

These transforms allow you to generate, map and merge rows from two or more sources, create SQL query operations (expressions, lookups, joins, and filters), perform conditional splitting, and mask personal data to keep sensitive data relevant, anonymous, and secure.

[Case \[page 833\]](#)

[Data Mask \[page 836\]](#)

The Data Mask transform enables you to protect personally identifiable information in your data

[DQM Microservices \[page 925\]](#)

With the DQM Microservices transform, you can configure and execute SAP Data Quality Management, microservices for location data services within SAP Data Services.

[Map_Operation \[page 938\]](#)

[Merge \[page 942\]](#)

[Query \[page 943\]](#)

The Query transform retrieves a data set that satisfies conditions that you specify. This transform is similar to a SQL `SELECT` statement.

[Row_Generation \[page 970\]](#)

[SQL \[page 972\]](#)

[User-Defined \[page 975\]](#)

[Validation \[page 983\]](#)

[XML_Map \[page 994\]](#)

8.6.1 Case



Specifies multiple paths in a single transform (different rows are processed in different ways).

The Case transform simplifies branch logic in data flows by consolidating case or decision making logic in one transform. Paths are defined in an expression table.

Parent topic: [Platform transforms \[page 833\]](#)

Related Information

[Data Mask \[page 836\]](#)

[DQM Microservices \[page 925\]](#)

[Map_Operation \[page 938\]](#)

[Merge \[page 942\]](#)

[Query \[page 943\]](#)

[Row_Generation \[page 970\]](#)

[SQL \[page 972\]](#)

[User-Defined \[page 975\]](#)

[Validation \[page 983\]](#)

[XML_Map \[page 994\]](#)

8.6.1.1 Data inputs

Only one data flow source is allowed.

8.6.1.2 Editor

The Case transform editor consists of a Schema In pane and a Case tab.

The Case tab includes:

- An expression table that lists labels and their CASE expressions
 - An embedded Smart Editor for the CASE expression
 - A [Functions](#) button that open the Function Wizard
 - An ellipses (...) button that opens a full-size Smart Editor
1. Use the buttons, or right-click the expression table to insert or delete cases.

While using this table, the window also allows you to:

- Select multiple rows
- Apply delete functionality to a multiple selection

- Press Delete or Insert keys to delete or add an expression
- 2. In the expression table, click a label to rename it.
- 3. Enter the expression in the editor (drag columns from the input schema).

For large expressions, open the smart editor. Both the smart editor and the function wizard can assist you with expression creation.

- 4. To add a DEFAULT case, select the *Produce default output when all expressions are false* check box. The label changes to read *Produce default output with label: default*. Change the label name if desired.

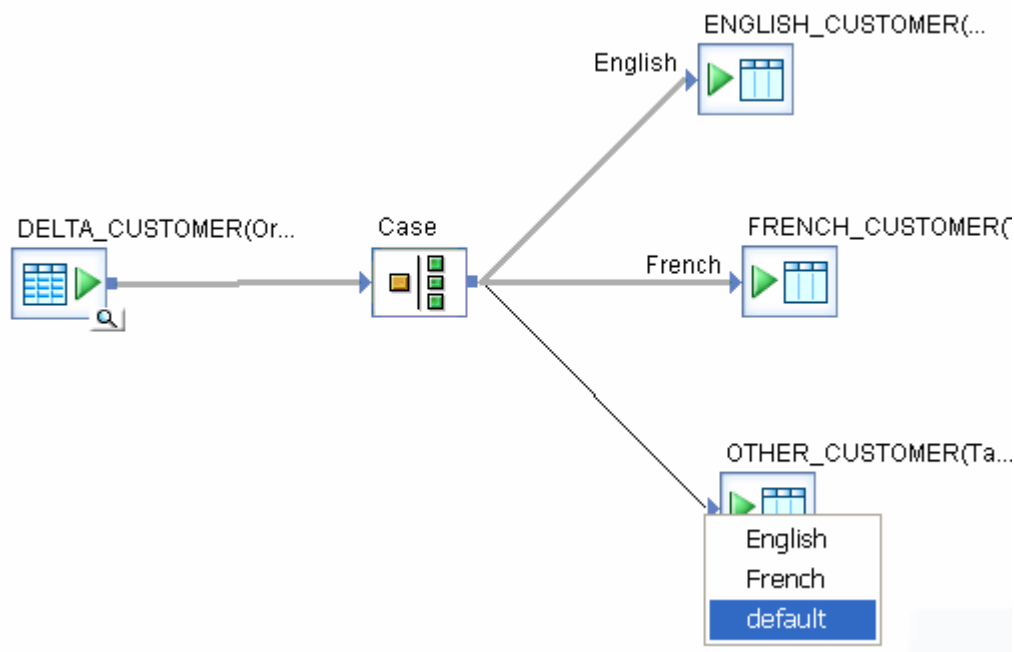
When you add a DEFAULT case, Data Services will send rows to this case when all other case conditions are false.

8.6.1.3 Options

Option	Description
<i>Expression</i>	<p>CASE expression for the corresponding label.</p> <p>DEFAULT is the expression used when all other CASE expressions evaluate to false. To enable DEFAULT, select the <i>Produce default output when all expressions are false</i> check box. The label changes to read <i>Produce default output with label: default</i>. Change the label name if desired.</p>
<i>Label</i>	<p>Name of the connection description indicating where data will go if the corresponding CASE condition is true.</p>
<i>Preserve expression order</i>	<p>This option is available only when the <i>Row can be TRUE for one case only</i> option is checked.</p> <p>Select this option if expression order is important to you.</p> <p>When this option is NOT checked, you can increase the performance of the Case transform, because Data Services will reorder your expressions to process them in a less CPU intensive manner first.</p> <p>This reordering of expressions can change your results, because there is no way to guarantee which expression will evaluate to TRUE first.</p>
<i>Produce default output with label</i> or <i>Produce default output when all expressions are false</i>	<p>This option changes depending on whether it is selected.</p> <p>Select <i>Produce default output with label</i> to send rows to this case when all other case conditions are false.</p>
<i>Row can be TRUE for one case only</i>	<p>If this option is selected, the row is passed to the first case whose expression returns TRUE. Otherwise, the row is passed to all the cases whose expression returns TRUE.</p> <p>For jobs created in the 6.0 release or earlier, this option is set to FALSE. When you create a case transform in 6.1or later, this option defaults to TRUE</p>

8.6.1.4 Data outputs

Connect the output of the Case transform with another object in the workspace. Choose a case label from a pop-up menu. Each label represents a case expression (WHERE clause) created in the Case editor.



The connections between the Case transform and objects used for a particular case must be labeled. Each output label in the Case transform must be used at least once.

To delete a case connection, right-click the connection label and select *Delete*.

The Case transform can be used to implement IF-THEN-ELSE logic rather than using a conditional flow. However:

- Conditionals operate at the work flow level
- Case transforms operate within data flows

8.6.2 Data Mask

The Data Mask transform enables you to protect personally identifiable information in your data



Personally identifiable information (PII) includes data such as credit card numbers, salary information, birth dates, personal identification numbers, or bank account numbers. You may want to use data masking to support security and privacy policies, and to protect your customer or employee information from possible theft or exploitation.

❖ Example

Mask employee salary information when you send your employee database to developers and testers in your company who do not have the proper security clearance to view such information, but who need realistic data to correctly develop and test the application.

Placement in data flow

Place the Data Mask transform toward the end of your data flow to ensure that all fields that are to be masked have undergone processing by upstream transforms. If you place the Data Mask transform before other transforms in your data flow, downstream transforms may not be processing actual data but masked data, and in some cases, the transform won't be able to process the fields at all if the Data Mask transform replaced input data with blanks or a mask-out character such as "#".

⚠ Caution

The data masking changes are irreversible. Therefore, ensure that you preserve the original databases before masking data.

SAP Data Services uses the following techniques for data masking:

- Mask out
- Number variance
- Date variance
- Pattern variance
- Number generalization
- Date generalization

Content objects

SAP Data Services Designer has a read-only copy of the Data Mask transform in the [Transform](#) tab. Find [Base_DataMask](#) under the [Platform](#) group of transforms, in the [Object Library](#).

You can set default options in a copy of the Base_DataMask transform. Set up default options in the Data Mask [Transform Configuration Editor](#) window, and also set best practice input fields, and best practice output fields for your transform configuration. After you place an instance of the transform configuration in a data flow, you can override your preset defaults if necessary.

⚠ Caution

If you edit a transform configuration that is used in more than one data flow, every instance of the transform configuration inherits that change, unless a user has explicitly overridden the same option value in an instance.

Parent topic: [Platform transforms \[page 833\]](#)

Related Information

[Case](#) [page 833]
[DQM Microservices](#) [page 925]
[Map_Operation](#) [page 938]
[Merge](#) [page 942]
[Query](#) [page 943]
[Row_Generation](#) [page 970]
[SQL](#) [page 972]
[User-Defined](#) [page 975]
[Validation](#) [page 983]
[XML_Map](#) [page 994]

8.6.2.1 Input fields for the Data Mask transform

Data mask works with input fields that contain personal data.

Data Mask input fields

Input field	Description
Date[1-6]	For example, 08/16/2013 or August 16 2013. Available for mask out, date variance, and date generalization.
Email[1-6]	E-mail address. Available for mask out and pattern variance.
Numeric_Data[1-6]	Numeric and character data. Supported numeric data types include decimal, double, integer, and real. Available for mask out, number variance, and number generalization.
Phone[1-6]	Phone number in any format such as a phone number that includes country code, area code, and formatting characters such as parentheses, dashes, and spaces. Available for mask out and pattern variance.
SSN[1-6]	U.S. Social Security number. Available for mask out and pattern variance.
General_Data[1-12]	Data that does not apply to any of the other Data Mask transform input fields. Available for mask out and pattern variance.

Select the *Best practice*, *In use*, or *All* for the *Filter* at the top of the *Input* tab to view the corresponding input fields.

Each of the masking groups in the Data Mask transform has an option named *Mapped Input Field*. To populate the drop list for this option, map your input fields to the appropriate transform input fields in the *Input* tab (*Transform Input Field Name* column). The following table lists the appropriate transform input fields for each data mask option group.

Option group	Map to transform input field name
Mask Out	<ul style="list-style-type: none"> • Date[1-6] • Email[1-6] • Numeric_Data[1-6] • Phone[1-6] • SSN[1-6] • GeneralData[1-12]
Number Variance	Numeric_Data[1-6]
Date Variance	Date[1-6]
Pattern Variance	<ul style="list-style-type: none"> • Email[1-6] • Phone[1-6] • SSN[1-6] • General_Data[1-12]
Number Generalization	Numeric_Data[1-6]
Date Generalization	Date[1-6]

Note

To maintain formatting characters (such as commas, periods, slashes, and spaces) in the output field for mask out data masking, set the input *Data Type* (in the *File Format Editor*) to varchar. Additionally, set the *Maintain Formatting* option (in the *Mask Out* group in the transform) to Yes.

Related Information

[Data types and output results for Maintain Formatting \[page 846\]](#)

[Mask Out Group \[page 843\]](#)

[Number Variance Group \[page 851\]](#)

[Date Variance Group \[page 865\]](#)

[Pattern Variance Group \[page 879\]](#)

[Number Generalization Group \[page 900\]](#)

[Date Generalization Group \[page 905\]](#)

8.6.2.2 Output fields for the Data Mask transform

The software lists the output fields that you see in the [Output](#) tab of the Data Mask transform based on the mapped input fields and the options that you set for each applicable Data Mask group.

Open the [Output](#) tab and check the box next to all of the fields that you included in the Data Mask transform's [Mask Out](#), [Number Variance](#), [Date Variance](#), [Pattern Variance](#), [Number Generalization](#), and [Date Generalization](#) groups.

i Note

If you do not check all of the boxes in the [Output](#) tab that are included in the [Data Mask](#) transform, the software issues a job execution warning when you execute the job.

If the fields that you plan to use for data mask are not listed in the [Output](#) tab, make sure that you have mapped them correctly in the [Input](#) tab, and that you have set up the corresponding [Mask Out](#), [Number Variance](#), [Date Variance](#), [Pattern Variance](#), [Number Generalization](#), or [Date Generalization](#) groups in the Data Mask transform.

Select the [Best practice](#), [In use](#), or [All](#) for the [Filter](#) at the top of the [Output](#) tab to view the corresponding output fields.

You need to consider the output field size when you prepare the Data Mask transform for any type of masking. The output field length may not accommodate the length of the masked value. If the masked output value has more characters than the number of characters set for the output field, the software truncates the output data.

8.6.2.3 Processing Options

Processing Options

Option	Description
Seed	<p>(Optional) An alpha and/or a numeric string. Set this option once when you want to maintain referential integrity each time you run the job. One seed value maintains referential integrity for the following variance types set up in the Data Mask transform: Number Variance, Date Variance, and Pattern Variance.</p> <p>Additionally, to retain referential integrity for subsequent jobs using this job setup, you have to use the same data and you cannot make changes to the Data Mask transform, specifically to the applicable variance types.</p>

Related Information

[Use Seed value for referential integrity \[page 878\]](#)

8.6.2.4 Date Parsing Options

Settings that you make in the [Date Parsing Options](#) group in the Data Mask transform determine how the software parses input dates when the dates are ambiguous.

The software uses these settings for Date Variance and Date Generalization operations in the Data Mask transform.

An input date is ambiguous for various reasons:

- When the input day of the month in numerals is 12 or less.
- When the input month name is the same in full or short form.
- When the input month is the same word in more than one language.
- When the input year is two digits and it is unclear what century to use.

The [Date Parsing Options](#) group is optional but it cannot be blank. Leave the default settings or change the settings based on your input data. Ensure that your settings are valid for your input data. If your settings are not valid, the software may output NULL. If the software does not recognize the input date because the date is outside of the acceptable range or it is invalid, it may also output NULL.

You cannot repeat this group.

Date Parsing Options

Option	Description
Default Date Format	<p>Specifies the order in which month, day, and year elements appear in the input string.</p> <p>The software uses this value only when the day, month, or year in the input string is ambiguous. Choose one of these formats:</p> <ul style="list-style-type: none">• Day_Month_Year• Month_Day_Year Default setting• Year_Day_Month• Year_Month_Day
Default Month Format	<p>Specifies the month format when it is text and not numeric. The software uses this setting to output the month when it cannot determine the output month format based on the input alone.</p> <ul style="list-style-type: none">• Full: Output the spelled-out version of the month name.• Short: Output the abbreviated form of the month name, when an abbreviated form exists. Default setting.

Option	Description
Default Language	<p>Specifies the language that the software should use when parsing ambiguous input dates that use text for the month. The software uses the set language for output. The default language is English.</p> <div> <p>i Note</p> <p>The software does not verify that the user-defined Default Language corresponds to the language of the input month.</p> </div>
Default Century Threshold	<p>Indicates whether a two-digit date is considered part of the 20th or 21st century.</p> <p>Enter a value from 0 to 99. The default value is 25.</p> <p>Specify a two-digit integer that represents the first year that a parsed two-digit year is considered part of the 21st century (20xx). All two-digit years greater than the specified integer are considered part of the 20th century (19xx).</p>

Related Information

[Date Variance Group \[page 865\]](#)

[Date Generalization Group \[page 905\]](#)

8.6.2.4.1 Date Parsing examples

Use these examples to help you understand how Date Parsing settings affect ambiguous input dates.

❖ Example

An input month of “Juni” may be ambiguous for language and for month format.

- The month “Juni” is ambiguous for language because it is the same spelling for both German and Norweigan.
- The month “Juni” is ambiguous for month format because it is both the full and short month format in German and Norweigan.

i Note

“Juni” is used in other languages too, we only use the German and Norweigan version for example purposes.

❖ Example

The software parses the following input date differently based on the *Default Date Format* setting.

- Input date: 10/01/2017

Default Date Format = *DAY_MONTH_YEAR*

Software parses the date as January 10, 2017

- Input date: 10/01/2017

Default Date Format = *MONTH_DAY_YEAR*

Software parses the date as October 1, 2017

❖ Example

In English, the software randomizes an input date of May 5, 2014 to a randomized output date of March 22, 2013. However, because May is ambiguous in determining if the output is full or short, the software relies on the *Default Month Format* setting to determine the output format for month.

- When the *Default Month Format* = *FULL*, the software outputs "March" for the month.
- When the *Default Month Format* = *SHORT*, the software outputs "Mar" for the month.

❖ Example

If you enter 11 for the *Default Century Threshold* option, the software processes all two-digit years of 11 or lower as part of the 21st century:

- 08 is considered 2008
- 11 is considered 2011
- 12 is considered 1912

Related Information

[Date Parsing Options \[page 841\]](#)

8.6.2.5 Mask Out Group

Set Mask Out options to mask portions of output fields.

To set up the *Mask Out Group* of options you select the field that you want to mask, the portions of the field to remain unmasked (from the start or the end), the number of characters to remain unmasked, and the replacement character for the masked portion of the field.

You may duplicate this group as many times as needed.

i Note

For the mask out process, the transform outputs all of the masked output fields as Varchar regardless of the input field type. Additionally, the transform converts any non-Varchar input field to Varchar before it processes the field for masking.

i Note

To maintain formatting characters (such as commas, periods, slashes, and spaces) in the masked out output field, set the input *Data Type* (in the *File Format Editor*) to Varchar. Additionally, set the *Maintain Formatting* option (in the *Mask Out* group in the transform) to YES.

Mask Out Group options

Option	Description
<i>Mapped Input Field</i>	Select the name of the mapped input field that you want to mask from the drop-down list. <div><h2>i Note</h2><p>The drop-down list includes only the fields that you have mapped to one of the applicable fields that are listed under the <i>Transform Input Field Name</i> column in the <i>Input</i> tab.</p></div>
<i>Starting Position</i>	Indicates the portion of the field that remains unmasked. <i>Start</i> : Keeps the specified number of characters unmasked from the start of the string. <i>End</i> : Keeps the specified number of characters unmasked from the end of the string.
<i>Unmasked Length</i>	The number of characters to keep unmasked from the start or end of the field based on the <i>Starting Position</i> setting.
<i>Masking Character</i>	The character that replaces the original characters in the string. For example, "#" or "*". <div><h2>i Note</h2><p>If you leave this option blank, the transform outputs only the portion of the field that remains unmasked. For example, with <i>Starting Position</i> set to END, <i>Unmasked Length</i> set to 4, and no setting for <i>Masking Character</i>, the number "123456789" is output as "6789". However, when you set the <i>Masking Character</i> option to the "#" character, the output is "#####6789".</p></div>

Option	Description
<i>Maintain Formatting</i> (for fields other than Email[1-6])	<p>Indicates whether to keep formatting in the output field.</p> <p>Yes: Keeps any spaces, punctuation, and field formatting, such as dashes, slashes, or periods, that are present in the input field. For example, XXX-XXX-1234.</p> <p>No: Treats any spaces between characters, punctuation, and field formatting, such as dashes, slashes, or periods, like all of the other characters in the string, and outputs them using the masking character that you designate. For example, the dashes are masked with "X" (shown bolded for this example):</p> <p>XXXXXXXX1234.</p> <div> <p>i Note</p> <p>To maintain formatting characters (such as commas, periods, slashes, and spaces) in the masked out output field, set the input <i>Data Type</i> (in the <i>File Format Editor</i>) to Varchar. Additionally, set this option, <i>Maintain Formatting</i>, to YES.</p> </div> <div> <p>i Note</p> <p>Maintain Formatting works differently if you are masking out an input field that is mapped to the transform input field Email[1-6]. See the <i>Maintain Formatting</i> (for Email[1-6] field only) row below for details.</p> </div>
<i>Maintain Formatting</i> (for Email[1-6] field only)	<p>Yes: Maintains the formatting in the domain portion of an email address (from the "@" character to the end of the email address). It treats any punctuation that comes before the "@" character (email name portion), as a part of the email name, and masks the punctuation along with the other characters in the string.</p> <div> <p>i Note</p> <p>The domain portion of an email address begins at the "@" character and includes formatting characters to the end of the address (the "@" and ".").</p> </div> <p>No: Works the same as for other fields. For Email[1-6] fields, it masks all of the formatting characters in the domain portion of the email address just like the formatting characters that appear in the name portion of the email address.</p>

Related Information

[Mask out examples \[page 847\]](#)

[Data types and output results for Maintain Formatting \[page 846\]](#)

[Data types and output results for Maintain Formatting \[page 846\]](#)

8.6.2.5.1 Data types and output results for Maintain Formatting

To maintain formatting characters on output for data masking, set the *Type* to Varchar. Make this setting in the *File Format Editor* for the specific field. Additionally, set the *Maintain Formatting* option in the *Mask Out* group to Yes.

❖ Example

The examples in the following table show how the software outputs formatting from input fields based on the input data type and the *Mask Out* settings.

Mask Out option settings:

- *Starting Position* = End
- *Unmasked Length* = 4
- *Masking Character* = X
- *Maintain Formatting* = Yes

Data types and output results for Maintain Formatting

Input	Data type	Output	Comment
2014.12.24	Date	XXXX1224	Software does not output the periods, even though the <i>Maintain Formatting</i> option is set to Yes, because the input data type is Date.
2014.12.24	Varchar	XXXX.12.24	Software maintains the periods on output because the data type is Varchar.
2014.12.24 21:30:00	Datetime	XXXXXXXXXX3000	Software does not output the periods, space, and colons, even though the <i>Maintain Formatting</i> option is set to Yes, because the input data type is Datetime.
2014.12.24 21:30:00	Varchar	XXXX.XX.XX XX:30:00	Software maintains the periods, space, and colons on output because the data type is Varchar.

Input	Data type	Output	Comment
1,234.56	Double	XX34.56	Software does not output the thousands separator, even though the Maintain Formatting option is set to Yes, because the data type is Double. i Note This example also applies to other numeric data types.
1,234.56	Varchar	X,X34.56	Software maintains the thousands separator and the period on output because the data type is Varchar.
1.234,56	Double	XX34.56	Software does not maintain the thousands separator, even though the Maintain Formatting option is set to Yes, because the data type is Double. Also, the software changes the input comma to a period on output. i Note This example also applies to other numeric data types.
1.234,56	Varchar	X.X34,56	Software maintains the period and comma on output because the data type is Varchar.

8.6.2.5.2 Mask out examples

The following tables illustrate how the [Mask Out Group](#) transform options process various types of information such as U. S. Social Security Numbers, phone numbers, dates, and general data.

i Note

To maintain formatting characters (such as commas, periods, slashes, and spaces) in the masked out output field, set the [Data Type](#) for the applicable input field to varchar in the [File Format Editor](#). Additionally, set the [Maintain Formatting](#) option in the [Mask Out](#) options to Yes.

For all of the examples below, the input field data type is set to varchar.

U. S. Social Security Numbers

Option settings	Original data	Output:	Output:
		Masking character = #	Masking character = None
<i>Starting Position</i> = End <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = (Yes or No)	123456789	#####6789	6789
<i>Starting Position</i> = End <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = Yes	123-45-6789	###-##-6789	6789
<div> <div>i Note</div> <div> <p>The transform maintains the dashes as formatting characters in the first output column because the <i>Maintain Formatting</i> option setting is Yes. When there is no masking character set as in the last output column, the transform does not maintain the formatting so the last column output does not include a dash.</p> </div> </div>			
<i>Starting Position</i> = End <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = No	123-45-6789	#####6789	6789
<i>Starting Position</i> = End <i>Unmasked Length</i> = 5 <i>Maintain Formatting</i> = Yes	123-45-6789	###-#5-6789	5-6789
<i>Starting Position</i> = End <i>Unmasked Length</i> = 5 <i>Maintain Formatting</i> = No	123-45-6789	#####-6789	-6789
<i>Starting Position</i> = Start <i>Unmasked Length</i> = 3 <i>Maintain Formatting</i> = Yes	123-45-6789	123-##-####	123
<i>Starting Position</i> = Start <i>Unmasked Length</i> = 3 <i>Maintain Formatting</i> = No	123-45-6789	123#####	123
<i>Starting Position</i> = Start <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = Yes	123 45 6789	123 4# #####	123 4

Option settings	Original data	Output:	Output:
		Masking character = #	Masking character = None
<i>Starting Position</i> = Start <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = No	123 45 6789	123 #####	123

Phone numbers

Option settings	Original data	Output:	Output:
		Masking character = #	Masking character = None
<i>Starting Position</i> = End <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = (Yes or No)	16085551212	#####1212	1212
<i>Starting Position</i> = End <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = Yes	1 (608) 555-1212	# (###) ###-1212	1212
<i>Starting Position</i> = End <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = No	1 (608) 555-1212	#####1212	1212
<i>Starting Position</i> = End <i>Unmasked Length</i> = 5 <i>Maintain Formatting</i> = Yes	555-1212	##5-1212	5-1212
<i>Starting Position</i> = End <i>Unmasked Length</i> = 5 <i>Maintain Formatting</i> = No	555-1212	###-1212	-1212

Dates

Option settings	Original data	Output:	Output:
		Masking character = #	Masking character = None
<i>Starting Position</i> = End <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = (Yes or No)	04142014	####2014	2014
<i>Starting Position</i> = End <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = Yes	04/14/2014	##/##/2014	2014
<i>Starting Position</i> = End <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = No	04/14/2014	#####2014	2014

Option settings	Original data	Output: Masking character = #	Output: Masking character = None
<i>Starting Position</i> = End <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = Yes	April 14, 2014	##### ##, 2014	2014
<i>Starting Position</i> = End <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = No	April 14, 2014	#####2014	2014
<i>Starting Position</i> = Start <i>Unmasked Length</i> = 5 <i>Maintain Formatting</i> = Yes	July 4, 2014	July 4, #####	July 4
<i>Starting Position</i> = Start <i>Unmasked Length</i> = 5 <i>Maintain Formatting</i> = No	July 4, 2014	July #####	July

Email addresses

Unlike other personally identifiable transform input fields (Date, Numeric_Data, Phone, SSN, and General_Data), the software considers only portions of an email address to have “formatting” characters.

There are two parts to an email address. Each part is separated by the “@” symbol:

- The portion to the left of the @ symbol is the email name portion
- The portion to the right of the @ symbol is the domain name.

Email

Option settings	Original data	Output: Masking character = #	Output: Masking character = None
<i>Starting Position</i> = End <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = Yes	some.one@sap.com	#####@##p.com	p.com
<i>Starting Position</i> = End <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = No	some.one@sap.com	#####.com	.com
<i>Starting Position</i> = Start <i>Unmasked Length</i> = 3 <i>Maintain Formatting</i> = Yes	some.one@sap.com	som#####@###.###	som
<i>Starting Position</i> = Start <i>Unmasked Length</i> = 3 <i>Maintain Formatting</i> = No	some.one@sap.com	som#####	som

General data

Option settings	Original data	Output:	Output:
		Masking character = #	Masking character = None
<i>Starting Position</i> = End <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = Yes	High & low.	###h & low.	h & low.
<i>Starting Position</i> = End <i>Unmasked Length</i> = 4 <i>Maintain Formatting</i> = No	High & low.	#####low.	low.
<i>Starting Position</i> = Start <i>Unmasked Length</i> = 2 <i>Maintain Formatting</i> = Yes	High & low.	Hi## & ###.	Hi
<i>Starting Position</i> = Start <i>Unmasked Length</i> = 2 <i>Maintain Formatting</i> = No	High & low.	Hi#####	Hi
<i>Starting Position</i> = End <i>Unmasked Length</i> = 3 <i>Maintain Formatting</i> = Yes	Jon Q. Boyd-Low	### #. #####-Low	-Low
<i>Starting Position</i> = End <i>Unmasked Length</i> = 2 <i>Maintain Formatting</i> = No	Jon Q. Boyd-Low	#####ow	ow

Related Information

[Data types and output results for Maintain Formatting \[page 846\]](#)

[Input fields for the Data Mask transform \[page 838\]](#)

[Mask Out Group \[page 843\]](#)

8.6.2.6 Number Variance Group

Alter numeric input values based on a variance type (fixed number, percentage, or range) and other settings in the [Number Variance Group](#). Keep referential integrity using a seed value to keep altered number values the same when you run a job multiple times.

You may duplicate this group of options as many times as needed to apply the number variance settings to different output fields.

Note

You can set up number variance on any type of input field (for example, varchar and integer). The transform processes any non-numeric fields as varchar by recognizing numeric data and varies the numeric data

based on settings in the transform. The transform retains the field type from input to output for all of the fields used for number variance.

Number Variance options

Option	Description
Mapped Input Field	<p>Select the name of the mapped input field that you want to output with number variance masking.</p> <div><p>i Note</p><p>The drop-down list includes only the fields that you have mapped to one of the Numeric_Data[1-6] fields that are listed in the Transform Input Field Name column in the Input tab.</p></div>
Type of Variance	<p>Percentage: Changes the input number to a random number that is within a determined minimum and maximum range (see Note). The transform determines the range based on input value, percentage value, and the user-set minimum and maximum values.</p> <p>Fixed Value: Changes the input number to a random number that is within a determined minimum and maximum range (see Note). The transform determines the range based on input value, fixed number value, and the user-set minimum and maximum values.</p> <div><p>i Note</p><p>For Percentage and Fixed Value, the software calculates internal minimum and maximum values. If the calculated minimum and maximum values do not violate (fall outside of) any user-set minimum and maximum values, the transform outputs a random number that is at or within the calculated minimum and maximum values. Sometimes the transform needs to use a combination of user-set and calculated minimum and maximum values.</p></div> <p>Range: Changes the input number to any number that is greater than or equal to the user-defined minimum value without going over the maximum value, regardless of the input value.</p> <div><p>i Note</p><p>The Variance option is not applicable for Range, and the transform ignores any value in Variance when you choose Range for this option.</p></div>

Option	Description
<i>Variance</i>	<p>Specify the variance number based on the <i>Type of Variance</i>: Percentage or fixed (not applicable for range). Variance must be greater than zero and can have decimal places.</p> <div> i Note <p>The <i>Variance</i> option is not applicable for the range variance type. The transform ignores any value in <i>Variance</i> when you select Range.</p> </div>
<i>Minimum Value</i>	<p>Required for range variance type and optional for percentage and fixed variance types. Specify the lowest value that the transform can output for the mapped input field. This number can be a decimal. For best results, always set a realistic minimum value. You may see some unrealistic numeric variations or null values if you do not set a realistic minimum value.</p> <div> i Note <p>The transform calculates an internal minimum value for each record. If the calculated minimum doesn't violate (is less than) the user-defined minimum set for this option, the transform bases the random output on the calculated minimum.</p> </div>
<i>Maximum Value</i>	<p>Required for range variance type and optional for percentage and fixed variance types. Specify the highest value that the transform can output for the mapped input field. This number can be a decimal. For best results, always set a realistic maximum value. You may see some unrealistic numeric variations or null values if you do not set a realistic maximum value.</p> <div> i Note <p>The transform calculates an internal maximum value for each record. If the calculated maximum doesn't violate (is greater than) the user-defined maximum set for this option, the transform bases the random output on the calculated maximum.</p> </div>
<div> i Note <p>Best practice is to always complete the <i>Minimum Value</i> and the <i>Maximum Value</i> options regardless of the variance type, and use realistic values to avoid unrealistic numeric values or null output.</p> </div>	

Related Information

[Calculated minimum and maximum values \[page 854\]](#)

[Number variance min/max examples \[page 855\]](#)

[Variance type: Percentage \[page 859\]](#)

[Variance type: Fixed number \[page 860\]](#)

[Variance type: Range \[page 862\]](#)

8.6.2.6.1 Calculated minimum and maximum values

The Data Mask transform uses internally-calculated minimum and maximum values to determine a random output value, except when they violate the user-set minimum and/or maximum values set in the transform.

If the calculated minimum and maximum values do not violate the user-set minimum and maximum values, the transform uses the calculated minimum and/or maximum for an output value, and outputs a random number that is equal to or within the calculated minimum and maximum values. Sometimes the determined minimum and maximum values are a combination of the user-set and the calculated minimum and maximum values. For example, if the calculated minimum value falls outside of the user-set minimum (calc. min. < set min.), the transform uses the user-set minimum value. If the calculated maximum falls outside of the user-set maximum (calc. max. > set max.), the transform uses the user-set maximum.

Calculated minimum and maximum for fixed variance type

The transform uses the following formula for calculated minimum and maximum values for fixed number variance:

$\text{input} - \text{variance} = \text{calc. min.}$

$\text{input} + \text{variance} = \text{calc. max.}$

For example, given an input value of 500, and a variance of 25, the calculated minimum and maximum formulas are:

Calculated minimum: $500 - 25 = 475$

Calculated maximum: $500 + 25 = 525$

Calculated minimum and maximum for percentage variance type

The transform uses the following formula for calculated minimum and maximum values for percentage number variance:

$\text{Input} - (\text{input} \times \text{variance \%}) = \text{calc. min.}$

$\text{Input} + (\text{input} \times \text{variance \%}) = \text{calc. max.}$

For example, given an input value of 500 and a variance percent of 25, the calculated minimum and maximum formulas are:

Calculated minimum: $500 - (500 \times 25\%) = 375$

Calculated maximum: $500 + (500 \times 25\%) = 625$

i Note

$(500 \times 25\% = 125)$

Related Information

[Number variance min/max examples \[page 855\]](#)

8.6.2.6.1.1 Number variance min/max examples

❖ Example

The examples below help to illustrate how the software calculates the number variance, especially when one or both of the user-set *Minimum Value* or *Maximum Value* fields are blank or fall outside of the calculated minimum and maximum values.

i Note

It is important to set minimum and maximum values that result in realistic variances. For example, in the following table there are some cases when the output is null, which may not be realistic for your purposes. The best practice is to always set a *Minimum Value* and *Maximum Value* in the transform options.

Input = 10

Type of Variance = Fixed Number

Variance = 5

Minimum Value = (as set in the User-Defined minimum column in the following table)

Maximum Value = (as set in the User-Defined maximum column in the following table)

Calculated minimum value: 5

Calculated maximum value: 15

Example	User minimum	User maximum	Minimum used	Maximum used	Notes
1	Not set	Not set	5 (calculated)	15 (calculated)	Output is a random number between the calculated range of 5-15. i Note There are no user minimum and maximum settings so the software uses the calculated minimum and maximum.
2	2 (under calculated minimum)	20 (over calculated maximum)	5 (calculated)	15 (calculated)	Output is a random number between the calculated range of 5-15. i Note The user-set minimum value falls outside of the calculated minimum. The user-set maximum also falls outside of the calculated maximum.

Example	User minimum	User maximum	Minimum used	Maximum used	Notes
3	Not set	4 (under calculated minimum)	--	--	Output is null. i Note There is no user-set minimum value, and the user-set maximum value falls outside of the 5-15 calculated range so there is no range for randomization.
4	Not set	5 (equal to calculated minimum)	--	--	Output is null. i Note There is no user-set minimum value and the user-set maximum value is equal to the calculated minimum. Therefore there is no range for randomization.
5	Not set	20 (over calculated maximum)	5 (calculated)	15 (calculated)	Output is a random number between the calculated range of 5-15. i Note There is no user-set minimum value, and the user-set maximum value falls outside of the calculated range.

Example	User minimum	User maximum	Minimum used	Maximum used	Notes
6	Not set	14 (under calculated maximum)	5 (calculated)	14 (user)	<p>Output is a random number between 5-14.</p> <div> <p>i Note</p> <p>There is no user-set minimum value. However, the user-set maximum value falls within the calculated maximum. The software does not violate the user-set maximum value because it is within the calculated range.</p> </div>
7	15 (equal to calculated maximum)	Not set	--	--	<p>Output is null.</p> <div> <p>i Note</p> <p>The user-set minimum value equals the calculated maximum value, and there is no user-set maximum value. Therefore, there is no range for randomization.</p> </div>

Example	User minimum	User maximum	Minimum used	Maximum used	Notes
8	20 (over calculated range)	Not set	--	--	Output is null.

i Note
The user-set minimum value falls outside of the calculated maximum value (and the calculated range), and there is no user-set maximum value. Therefore there is no range for randomization.

8.6.2.6.2 Variance type: Percentage

The transform outputs a random number that is within a determined range. It determines the range by calculating minimum and maximum values based on the input value, percentage value, and the minimum and maximum values that you set in the job.

Because the transform considers both calculated and user-set minimum and maximum values when determining an output range, it is important that you complete the [Minimum Value](#) and [Maximum Value](#) options when you set up your job (even though they are optional) with realistic values. You may see some unrealistic numeric variations or null values if you do not set realistic minimum and maximum values.

❖ Example

Number Variance option settings:

[Mapped Input Field](#) = Income

[Type of Variance](#) = Percentage

[Variance](#) = 25

i Note

The Variance must be greater than zero and it can be a decimal.

[Minimum Value](#) = 100,000

[Maximum Value](#) = 500,000

Calculated minimum = input - (input x 25%)

Calculated maximum = input + (input x 25%)

Record	Input	Calculated min/max	Output
1	250,000	187,500 and 312,500	<p>The output is a random number from 187,500 to 312,500.</p> <p>i Note</p> <p>The transform uses the calc. min. because it is greater than the user-set min. of 100,000. The transform uses the calc. max. because it is less than the user-set max. of 500,000.</p>
2	95,000	71,250 and 118,750	<p>The output is a random number between 100,000 and 118,750.</p> <p>i Note</p> <p>The transform uses a mixture of the user-set minimum and the calculated maximum. The calc. min. of 71,250, violates (is less than) the user-set min. of 100,000. The calc. max. of 118,750 does not violate (is less than) the user-set max. of 500,000.</p>

Related Information

[Calculated minimum and maximum values \[page 854\]](#)

[Number variance min/max examples \[page 855\]](#)

[Number formats \[page 863\]](#)

8.6.2.6.3 Variance type: Fixed number

The transform outputs a random number that is within a determined range. It determines the minimum and maximum range values to use based on the input value, fixed number value, and the minimum and maximum range that you set in the job.

Because the transform considers both calculated and user-set minimum and maximum values when determining an output range, it is important that you complete the *Minimum Value* and *Maximum Value* options when you set up your job (even though they are optional) with realistic values. You may see some unrealistic numeric variations or null values if you do not set realistic minimum and maximum values.

❖ Example

Number Variance option settings:

Type of Variance = Fixed Number

Mapped Input Field = monthly_income

Variance = 2500

i Note

You must set the variance to greater than zero, and you can use decimal values.

Minimum Value = 1,000

Maximum Value = 5,000

For this example, the mapped input field contains a phrase consisting of words, spaces, and multiple number groupings. When your mapped input field contains alphanumeric data like this, the transform applies the fixed number variance to only the first group of numbers in the field, and does not change any subsequent groups of numbers in the field. Therefore, the second number group, "2014", remains unchanged.

Record	Input	Calculated min/max	Output
1	Income/month \$1200 for 2014	-1,300 and 3,700	The output string is a random number from 1000 to 3,700:

i Note

The transform uses the user-set min. of 1000 because the calc. min. is greater than the set min. of -1,300. The transform uses the calc. max. because it is less than the set max of 5,000.

Record	Input	Calculated min/max	Output
2	Income/month \$2500 for 2014	0 and 5,000	The output string is a random number from 1,000 to 5,000.

i Note
The transform uses the user-set min because the calc. min. of 0 is less than the set. min. The cal. max. and user-set max are equal so the value 5,000 is used for the max.

❖ Example

If the input phrase lists “2014” as the first number group, the transform randomizes “2014” in all records, leaving the monthly income unmasked. Make sure you consider your input when setting up number variance.

For the following example, we used the same Number Variance option settings as in the first example:

Input	Calculated min/max	Output
“Income/month for 2014 is \$1200”	-486 and 4,514 (calculated using 2014 as the input number)	“Income/month for 3527 is \$1200”

i Note
The transform uses the set min. of 1,000 because the calc.min of -486 is less than the set min. The transform uses the calc. max. of 4,514 because it is less than the set max. of 5,000.

Related Information

[Calculated minimum and maximum values \[page 854\]](#)

[Number variance min/max examples \[page 855\]](#)

[Number formats \[page 863\]](#)

8.6.2.6.4 Variance type: Range

The transform randomly changes the input number to any number that is greater than or equal to the *Minimum Value* without going over the *Maximum Value*, regardless of the input value. The transform does not use a

calculated minimum or maximum value for range variance type. The transform ignores any value in the *Variance* option.

❖ Example

Number Variance option settings:

Mapped Input Field = Income

Type of Variance = Range

Variance = (not set)

Minimum Value = 100,000

Maximum Value = 500,000

Record	Input	Output
1	45,000	A random number between 100,000 and 500,00
2	525,000	A random number between 100,000 and 500,00

8.6.2.6.5 Number formats

The Data Mask transform outputs several types of number indicators when they are present in the input field:

- Thousand separators (comma, period, and space)
- Decimal separator (comma and period)
- Sign indicator (+/-)
- Number of significant figures

However, even when the field contains numbers and number formatting, the transform may not always recognize the entire input string as a number. The following table contains some examples of how the transform may vary numbers based on the format of the input number.

Examples: Number formats

Input data	Portion that transform recognizes as number (internal process)	Example of random output with number indicators from input data
1000000	1000000	2175924
1,000,000	1000000	2,175,924
1.000.000	1000000	2.175.924
1 000 000	1000000	2 175 924
10000	10000	25831
10,000	10000	25,831
10 000	10000	25 831
10.000	10.000	9.024

Input data	Portion that transform recognizes as number (internal process)	Example of random output with number indicators from input data
10	10	16
<div> i Note Notice the difference between the output for the previous row (9.024) and the output for this row (16). Even though the input numbers are both ten, the transform assigned the output in the previous row as a decimal because there is a decimal in the input value. </div>		
-15,2	-15.2	-1,9
+775	775	+801
-,9823	-0.9823	1,4761
+,.23400	0.23400	+1.23456
<div> i Note The transform adds a zero before the decimal before calculating the number variance output. </div>		
10,01,560	10.01	65,29,560
<div> i Note The transform recognizes the first number in the string (10,01) as a decimal. The transform does not modify the ",560" portion of the string because it doesn't match the pattern of a decimal. </div>		
9 10 11	9	26 10 11
<div> i Note The transform recognizes the first number in the string (9). The transform does not modify the "10" and "11" portions of the string. </div>		

Input data	Portion that transform recognizes as number (internal process)	Example of random output with number indicators from input data
100,250 25	100250	201,873 25
<div>i Note</div> <p>The transform recognizes the first number in the string (201,873). However, because there is a space after the first number, the transform does not modify the "25" portion of the string.</p>		
-123	-0.123	-0.531
<div>i Note</div> <p>The transform adds a zero ("0") before the decimal before it calculates the number variance output.</p>		

8.6.2.7 Date Variance Group

Use the [Date Variance Group](#) options to output randomized dates.

Set options in the [Date Variance Group](#) to alter input dates based on a date variance type (set number of days, months, years, or by date range) and other settings in the [Date Variance Group](#). Retain referential integrity using a seed value to keep the altered date values the same when you run a job multiple times.

Date Variance for data masking works with input fields that have been mapped to one of the Date[1-6] fields that are listed in the [Transform Input Field Name](#) column in the [Input](#) tab. The transform parses and then randomizes the date data based on the date variance settings in the transform. The transform retains the field type from input to output for all of the fields used for date variance.

You may duplicate this group of options as many times as needed to apply the date variance settings to different input fields.

Date Variance options

Option	Description
Mapped Input Field	Name of the input field that you want to output with date variance masking.
<div>i Note</div> <p>The drop-down list includes only the fields that you have mapped to one of the Date[1-6] fields that are listed in the Transform Input Field Name column in the Input tab.</p>	

Option	Description
<i>Type of Variance</i>	<p>Specifies how you want to vary a date.</p> <ul style="list-style-type: none"> • <i>Fixed Days</i>: Specifies to randomize the input date to somewhere within plus or minus a fixed number of days. • <i>Fixed Months</i>: Specifies to randomize a date to somewhere within plus or minus a fixed number of months. • <i>Fixed Years</i>: Specifies to randomize a date to somewhere within plus or minus a fixed number of years. <div> <p>i Note</p> <p>For Fixed Days, Fixed Months, and Fixed Years, the software calculates internal minimum and maximum dates. If the calculated minimum and maximum dates do not violate (fall outside of) any user-set minimum and maximum dates, the transform outputs a random date that is at or within the calculated minimum and maximum dates. Sometimes the transform needs to use a combination of user-set and calculated minimum and maximum dates.</p> </div> <ul style="list-style-type: none"> • <i>Range</i>: Specifies to randomize the input date to a date that is at or within a set minimum and maximum date range. You must complete the <i>Minimum Date</i> and <i>Maximum Date</i> options when you choose the Range variance type. <div> <p>i Note</p> <p><i>Variance</i> option is not applicable for Range, and the transform ignores any value in <i>Variance</i> when you choose Range for this option.</p> </div>
<i>Variance</i>	<p>Required only for the fixed variance types (days, months, years). Determines the number (days, months, years) by which to randomize the input. The value must be greater than zero (0).The</p> <div> <p>i Note</p> <p>If you choose the range variance type, the software ignores any value in this option.</p> </div>

Option	Description
Minimum Date	<p>Required for range variance and optional (but recommended) for fixed date variance types. Specify the minimum date allowed on output. Either type in a date in the YYYY.MM.DD format, or click the drop-down arrow in the field and click <Select Date...> to use the Set Date dialog box to set a date.</p> <p>The minimum date that can be set is September 14, 1752 (1752.09.14).</p> <div> <p>i Note</p> <p>The transform calculates an internal minimum date for each record. If the calculated minimum date doesn't violate (is less than or earlier than) the user-defined minimum set for this option, the transform bases the random output on the calculated minimum.</p> </div>
Maximum Date	<p>Required for range variance type and optional (but recommended) for fixed variance types. Specify the maximum date allowed on output. Either type in a date in the YYYY.MM.DD format, or click the drop-down arrow in the field and click <Select Date...> to use the Set Date dialog box to set a date.</p> <p>The maximum date that can be set is December 31, 9999 (9999.12.31).</p> <div> <p>i Note</p> <p>The transform calculates an internal maximum date for each record. If the calculated maximum date doesn't violate (is greater than or later than) the user-defined maximum setting for this option, the transform bases the random output on the calculated maximum.</p> </div>

Related Information

[Date Parsing Options \[page 841\]](#)

[Input fields for date variance \[page 868\]](#)

8.6.2.7.1 Input fields for date variance

Date variance for data masking works with date or varchar data types that you have mapped to a Date[1-6] field listed under the *Transform Input Field Name* column in the *Input* tab.

Date variance input field

Input data type	Date variance processing
Date	<p>The software varies the date based on your <i>date variance</i> settings in the job.</p> <p>The software has minimum and maximum date limits for date data types (these dates are different than the limits for character data types):</p> <ul style="list-style-type: none">• Minimum = September 14, 1752• Maximum = December 31, 9999 <p>If the input or randomized date falls outside of this range, or it is an invalid date for other reasons, then the software outputs null for the record.</p>
Datetime	<p>The software varies the date portion of the input field based on your <i>Date Variance</i> settings in the job. The software outputs the time portion of the input field unaltered.</p> <p>The minimum and maximum date limits for datetime are the same as for Date data types.</p> <p>If the input or randomized date falls outside of this range, or it is an invalid date for other reasons, then the software outputs null for the record.</p>

Input data type	Date variance processing
Character (like Varchar)	<p data-bbox="805 344 1385 432">The software parses and randomizes the date portion of a character string and outputs the randomized date. The output format is the same as it is on input.</p> <p data-bbox="805 456 1385 584">If there are two dates in a field, or there are other characters in the field, the software parses and randomizes the first date and outputs the other characters in the field (including the second date) unchanged.</p> <p data-bbox="805 609 1385 817">For example, if the input character string is 2013.05.01 - 2013.05.05, the software parses and randomizes the first date, and outputs the rest of the string unchanged. The output could be a character string of 2013.05.07 - 2013.05.05 (the software varies the first date based on the <i>Date Variance</i> settings in the job).</p> <p data-bbox="805 842 1385 936">The software has minimum and maximum date limits for character data type (these dates are different than the limits for date data types):</p> <ul data-bbox="815 960 1166 1025" style="list-style-type: none"> • Minimum is January 1, 1900 • Maximum is December 31, 2099 <p data-bbox="805 1050 1385 1144">If the software does not recognize the date (because it is outside of the acceptable range, or it is an invalid date for other reasons), then it will output null for the record.</p>

Related Information

[Descriptions of data types \[page 305\]](#)

[Input fields for the Data Mask transform \[page 838\]](#)

[Date Parsing Options \[page 841\]](#)

8.6.2.7.2 Calculated minimum and maximum dates

The software uses date variance settings in your job to calculate minimum and maximum dates to help determine a random output date.

The software also considers the user-set minimum and maximum settings from the job. The software uses the calculated minimum and/or maximum dates only when the user-set minimum and maximum dates fall outside of (violate) the calculated range.

Sometimes the minimum and maximum dates that the software uses are a combination of the user-set and the calculated minimum or maximum dates.

Related Information

[Examples: Calculated values for fixed options \[page 870\]](#)

8.6.2.7.2.1 Examples: Calculated values for fixed options

The following examples show how the software calculates minimum and maximum values for fixed days, months, and years.

❖ Example

Fixed days date variance

Given an input value of **2014.01.29** and a variance of 10 days, the calculated minimum and maximum dates are:

- Calculated minimum: 2014.01.29 - 10 days = 2014.01.19
- Calculated maximum: 2014.01.29 + 10 days = 2014.02.08

❖ Example

Fixed months date variance

Given an input value of **2014.01.29** and a variance of 10 months, the calculated minimum and maximum dates are:

- Calculated minimum: 2014.01.29 - 10 months = 2013.03.29
- Calculated maximum: 2014.01.29 + 10 months = 2014.11.29

❖ Example

Fixed years date variance

Given an input value of **2014.01.29** and a variance of 10 years, the calculated minimum and maximum dates are:

- Calculated minimum: 2014.01.29 - 10 years = 2004.01.29
- Calculated maximum: 2014.01.29 + 10 years = 2024.01.29

8.6.2.7.3 Date variance examples

❖ Example

The following table features some examples for one record in a database. The examples help to illustrate how the software calculates the date variance when one or both of the user-set *Minimum Date* or *Maximum Date* fields are blank or set to unrealistic date ranges.

i Note

It is important to set minimum and maximum values that result in realistic variances. For example, in the following table there are some cases when the output is null, which may not be realistic for your purposes. The best practice is to always set a realistic *Minimum Date* and *Maximum Date* in the transform options.

i Note

The input date format is YYYY.MM.DD.

- *Input Date* = 2014.09.17
- *Variance Type* = Fixed Days
- *Variance* = 7
- *Minimum Date* = (as set in the User minimum column in the following table)
- *Maximum Date* = (as set in the User maximum column in the following table)
- Calculated minimum date: 2014.09.10
- Calculated maximum date: 2014.09.24

Example	User minimum	User maximum	Minimum used	Maximum used	Notes
1	Not set	Not set	2014.09.10 (calculated)	2014.09.24 (calculated)	Output is a random date somewhere between the calculated minimum and maximum dates.

i Note

Because there are no user-set minimum and maximum dates, the transform uses the calculated minimum and calculated maximum dates to keep the randomized date within the transform settings.

Example	User minimum	User maximum	Minimum used	Maximum used	Notes
2	Not set	2014.09.09 (under the calculated minimum)	--	--	Output is null. i Note There is no user-set minimum date, and the user-set maximum date falls outside of the calculated date range of 2014.09.10 and 2014.09.24. There is no range for randomization.
3	Not set	2014.09.10 (equal to the calculated minimum range)	--	--	Output is null. i Note There is no user-set minimum date, and the user-set maximum date is equal to the calculated minimum date of 2014.09.10. There is no range for randomization.

Example	User minimum	User maximum	Minimum used	Maximum used	Notes
4	Not set	2014.09.23 (within the calculated range)	2014.09.10 (calculated)	2014.09.23 (user)	<p>Output is a random date somewhere between the calculated and user-set dates.</p> <p>i Note</p> <p>The user-set minimum is not set. The user-set maximum date is within the calculated range, and it is less than (earlier than) the calculated maximum date of 2014.09.24. The software won't violate the user-set maximum date.</p>
5	Not set	2014.09.24 (equal to calculated maximum)	2014.09.10 (calculated)	2014.09.24 (user)	<p>Output is a random date somewhere between the calculated and user-set dates.</p> <p>i Note</p> <p>The software uses the calculated minimum date because the user did not set a minimum date. It uses the user-set maximum date because it is equal to the calculated maximum date, and therefore does not violate it.</p>

Example	User minimum	User maximum	Minimum used	Maximum used	Notes
6	Not set	2014.09.30 (over calculated maximum)	2014.09.10 (calculated)	2014.09.24 (calculated)	Output is a random date somewhere between the calculated minimum and calculated maximum dates. <div> i Note There is no user-set minimum date so the software uses the calculated minimum date. The user-set maximum date is greater than (later than) the calculated maximum date. The software won't violate (go later than) the calculated maximum, so it uses the calculated maximum date. </div>

Example	User minimum	User maximum	Minimum used	Maximum used	Notes
7	2014.07.01 (under calculated minimum range)	Not set	2014.09.10 (calculated)	2014.09.24 (calculated)	Output is a random date somewhere between the calculated minimum of 2014.09.10 and the calculated maximum of 2014.09.24.
					<p>i Note</p> <p>The user-set minimum date is less than (earlier than) the calculated minimum date, and there is no user-set maximum date. Therefore the software uses the calculated minimum date because the user-set minimum violates the calculated date range, and it uses the calculated maximum because the user did not set a maximum date.</p>

Example	User minimum	User maximum	Minimum used	Maximum used	Notes
8	2014.07.01	2014.07.31	2014.07.01 (user)	2014.07.31 (user)	<p>A random number somewhere between the user-set minimum and maximum dates.</p> <p>i Note</p> <p>The user-set date range is totally outside of the calculated range of 2014.09.10 and 2014.09.24. Therefore, the software uses the user-set date range for randomization.</p>
9	2014.09.24 (equal to calculated maximum range)	Not set	--	--	<p>Output is null.</p> <p>i Note</p> <p>The user-set minimum date is equal to the calculated maximum date, and there is no user-set maximum date. There is no range for randomization.</p>

Example	User minimum	User maximum	Minimum used	Maximum used	Notes
10	2014.09.30 (greater than calculated maximum)	Not set	--	--	Output is null.

i Note

The user-set **minimum** date is greater than (later than) the calculated **maximum** date, and there is no user-set maximum date. There is no range for randomization.

Example	User minimum	User maximum	Minimum used	Maximum used	Notes
11	2014.09.11 (later than calculated minimum)	2014.09.30 (later than calculated maximum)	2014.09.11 (user)	2014.09.24 (calculated)	<p>The output is a random number somewhere between the user-set minimum date and the calculated maximum date.</p> <div> <p>i Note</p> <p>The user-set minimum date is greater than (later than) the calculated minimum date of 2014.09.10, but still less than (earlier than) the calculated maximum of 2014.09.24, so the user-set minimum is still within the calculated range. The user-set maximum date is greater than (later than) the calculated maximum date. Therefore the user maximum violates the calculated date range.</p> </div>

8.6.2.8 Use Seed value for referential integrity

Use the Seed value to make sure the software randomizes a field the same way from run to run. If you have set up the Data Mask transform so that a primary key field is randomized, and the primary key field is the only way you have to uniquely identify each record in the database, it is important that the software outputs the same randomized value for that primary key field for each time you process the database.

After you have run the job the first time, you can make job file adjustments for the second and subsequent runs, except you must always keep the [Seed](#) value, the number variance, date variance, or pattern variance options, and the specified variance input field values in the database the same from run to run.

The best practice for setting a seed value is to use a name or number or a combination of the two that you will easily remember. For example, if you are using the `<East_Sales_Region>` database, and the data is from 2014 sales, you may want to set a seed value of `eastsales2014`.

i Note

The [Seed](#) option applies only to the [Number Variance Group](#), [Date Variance Group](#), and the [Pattern Variance Group](#) in the [Data Mask](#) transform. You can include only one [Seed](#) value per transform. Therefore, if you used more than one [Number Variance](#), [Date Variance](#) and/or [Pattern Variance](#) group (which means you are randomizing more than just one input field), the set [Seed](#) value applies to all groups, so all fields that you include in the variance groups are randomized the same from run to run.

Related Information

[Number Variance Group \[page 851\]](#)

[Date Variance Group \[page 865\]](#)

8.6.2.9 Pattern Variance Group

Use the [Pattern Variance Group](#) to mask an input field substring with a specific pattern variance type.

Set options in the [Pattern Variance Group](#) to mask input data based on a pattern variance type (default, preserve, character, or string). Retain referential integrity using a seed value to keep the masked values the same when you run a job multiple times.

Pattern variance for data masking works with varchar input fields that are mapped to the following transform input fields: Email[1-6], Phone[1-6], SSN[1-6], and General_Data[1-12].

The transform parses and then randomizes an input field based on your settings in the [Pattern Variance Group](#).

The [Pattern Variance Group](#) consists of a [Pattern Variance](#) group. The [Pattern Variance](#) group has a [Definition](#) subgroup.

Duplicate the [Pattern Variance](#) group for each mapped input field that you want to define. Complete at least one [Definition](#) group per [Pattern Variance](#) group to define a substring and set pattern variance options. You can duplicate the [Definition](#) group to define additional substrings of the input field.

i Note

When you include multiple [Definition](#) groups for a mapped input field, each group must specify a unique substring of the input field. The software issues a verification error if you have defined overlapping substrings for the specified input field.

Pattern Variance Group subgroups and options

Group/Option	Description
<i>Pattern Variance</i> (duplicate for each input field that you want to mask)	
<i>Mapped Input Field</i>	Select the mapped input field for pattern variance from the drop list. The drop list includes all varchar input fields that you have mapped to the following transform input fields: Email[1-6], Phone[1-6], SSN[1-6], or General_Data[1-12].
<i>Definition</i> (duplicate for each unique substring of the mapped input field)	
<i>Starting Position</i>	<p>Specify the starting position for the substring. The software includes each alpha, numeric, space, and other printable character (special characters such as @, #, _, &) in the position count.</p> <p>The default starting position is 1.</p> <p>The software uses the <i>Starting Position</i> setting and the <i>Length</i> setting to identify the substring.</p>
<i>Length</i>	<p>Specify the number of positions (characters) to include in the substring.</p> <p>The default length is blank.</p> <p>The software uses the <i>Length</i> setting and the <i>Starting Position</i> setting to identify the substring.</p> <p>Leave the <i>Length</i> field blank to randomize the mapped input field from the set <i>Starting Position</i> to the end of the string.</p> <p>For example, set the <i>Length</i> to 2 for a two-character substring. If the <i>Starting Position</i> is set to 5, the substring consists of position five and six of the specified input field. Leave the <i>Length</i> blank in this example to mask all positions starting at position five to the end of the string.</p>
<i>Type of Variance</i>	<p>Specify the pattern variance type to apply to the defined substring of the <i>Mapped Input Field</i>. Choose one of the following pattern variance types:</p> <ul style="list-style-type: none"> • <i>Preserve</i>. Outputs the defined substring the same as it is input. • <i>Character</i>. Masks the defined substring by randomly replacing each of the characters in the defined substring with values that you specify in the <i>Value</i> field. Retains spaces and special input characters in the output field. • <i>String</i>. Masks the defined substring by randomly replacing the entire substring with values that you specify in the <i>Value</i> field. Does not retain spaces or special input characters in the output field. • <i>Default</i>. Masks each applicable character with like characters for alpha and numeric content. Retains any special input characters and spaces in the output field.

Group/Option	Description
<i>Value</i>	<div> <div>i Note</div> <div>You may leave this field blank for the preserve and the default pattern variance types. The software issues a validation error if you leave this field blank for any other pattern variance type.</div> </div> <p>Specify alpha and numeric characters, spaces, and special characters for masking the substring. The values you enter must comply with the pattern variance type you choose. For example, when you choose the string pattern variance type, enter alpha or numeric strings or numeric ranges. String pattern variance does not accept alphabetic ranges.</p> <p>See each pattern variance type section for more information about what you can include in the <i>Value</i> field.</p> <p>When you include more than one value, separate each value with a pipe delimiter (). For example, a value for the string pattern variance could include several characters that spell a word or an acronym; the entire string is considered one value. If you include another value after the first value, enter a pipe character () between the values to create a pipe-delimited list. For example:</p> <p>BMI BP HR</p>

Related Information

[Input fields for pattern variance \[page 881\]](#)
[Variance type: Default \[page 882\]](#)
[Variance type: Preserve \[page 884\]](#)
[Variance type: Character \[page 885\]](#)
[Variance type: String \[page 887\]](#)
[List multiple values for character and string pattern variance \[page 892\]](#)

8.6.2.9.1 Input fields for pattern variance

Pattern variance input fields must be varchar, and you must map them to the applicable transform input fields.

During job set up, you map varchar input fields to specific transform fields that are listed under the *Transform Input Field Name* column in the *Input* tab. Applicable transform fields for pattern variance are:

- Email[1-6]

- Phone[1-6]
- SSN[1-6]
- General_Data[1-12]

If you do not perform this mapping, the software does not list the field in the *Mapped Input Field* option under *Pattern Variance* in the *Options* tab.

i Note

If you want to mask date or numeric data types, you must use either mask out, date variance, or number variance instead of pattern variance.

Related Information

[Input fields for the Data Mask transform \[page 838\]](#)

8.6.2.9.2 Variance type: Default

The software masks each character with like characters for alpha and numeric content and retains any spaces and special characters (such as @, #, _, &) in the output field.

When you set up a *Definition* group for default pattern variance, you leave the *Value* field blank. The software ignores any values that you enter in the *Value* field.

The following table describes how the default pattern variance masks input characters with like characters.

Default pattern variance

Input character	Mask values
Alphabetic	<ul style="list-style-type: none"> • Masks lower-case alpha character with random lower-case alpha character • Masks upper-case alpha character with random upper-case alpha character
Numeric	Masks each numeric character with a random numeric character from 0 up to and including 9.
<ul style="list-style-type: none"> • Special character • Space 	Does not mask special characters or spaces, but outputs them as they are input, unmasked. For example, when the input substring contains a dash (-), the default pattern variance keeps the dash in the output. When the input substring contains a space, the default pattern variance keeps the space in the output.

The following table describes best practices for using the default pattern variance.

Best practices

Description	Best practice	Example Definition options
Mask an entire input field using the default pattern variance.	Accept the default settings in the <i>Definition</i> group for a mapped input field.	<p>Default <i>Definition</i> group settings:</p> <p><i>Starting Position</i> = 1</p> <p><i>Length</i> = <blank></p> <p><i>Type of Variance</i> = Default</p> <p><i>Value</i> = <blank></p>
Automatically apply the default pattern variance to substrings of a mapped input field that are not defined.	Define input field substrings using one or more of the other pattern variance types, and leave portions of the input field undefined.	<p>Definition 1:</p> <p><i>Starting Position</i> = 1</p> <p><i>Length</i> = 3</p> <p><i>Type of Variance</i> = Preserve</p> <p><i>Value</i> = <blank></p> <p>Definition 2:</p> <p><i>Starting Position</i> = 4</p> <p><i>Length</i> = 2</p> <p><i>Type of Variance</i> = String</p> <p><i>Value</i> = 20-25 50-55 90-95</p> <p>Undefined: Position six to the end of the field.</p> <p>Results</p> <p>Definition 1: The software preserves position one, two and three.</p> <p>Definition 2: The software masks the entire substring (position four and five) with a random number that is included in the ranges specified in the <i>Value</i> field (20-25, 50-55, or 90-95).</p> <p>Undefined: The software automatically masks position six to the end of the field with the default pattern variance.</p>

Related Information

[Pattern Variance Group \[page 879\]](#)
[Variance type: Preserve \[page 884\]](#)
[Variance type: Character \[page 885\]](#)
[Variance type: String \[page 887\]](#)
[Pattern variance examples \[page 897\]](#)

8.6.2.9.3 Variance type: Preserve

The software outputs the defined substring as it was input, with no masking.

i Note

The software applies the default pattern variance to any undefined portions of the input field. Undefined portions are the sections of the input field that have not been defined with preserve, character, or string pattern variance.

The following table contains an example of the preserve pattern variance. For more examples see [Pattern variance examples \[page 897\]](#).

❖ Example

The values listed in the **Input/Possible output** column in the following table show the masked substring for the preserve pattern variance in bold text.

Preserve pattern variance

Strategy	Definition options	Input/Possible output	Notes
Preserve the unit identification number in each record. Mask the rest of the field with the default pattern variance.	Undefined: Character one is not defined. Definition: <i>Starting Position</i> = 2 <i>Length</i> = 2 <i>Type of Variance</i> = Preserve <i>Value</i> = <ignored>	Input: A12:DER The following are possible output values: R 12 :KQG N 12 :UEP T 12 :RSL	Undefined: The software automatically masks the first position with a like character using the default pattern variance. Definition: The software preserves position two and three with the preserve pattern variance. Undefined: The software automatically masks the fourth position to the end of the string using the default pattern variance.
User sets up one <i>Definition</i> group.	Undefined: Character three to the end of the field are not defined.		<ul style="list-style-type: none">The colon in the input field (character four) is included in the undefined portion. The software outputs the colon as it was input based on the default pattern variance definition.

Related Information

[Pattern Variance Group \[page 879\]](#)

[Variance type: Character \[page 885\]](#)

[Variance type: String \[page 887\]](#)

[Variance type: Default \[page 882\]](#)

8.6.2.9.4 Variance type: Character

The software masks each character in the defined substring with a character from the [Value](#) field.

The [Value](#) field can include individual upper or lower case alpha characters, numeric characters from 0 to 9, ranges of alpha characters, or ranges of numeric characters (using numbers from 0 to 9), spaces, special characters (such as @, #, _, &), or any combination of these in a pipe-delimited list.

Note

Each alpha and numeric value must be one character in length. Numerics must be 0 to 9, including ranges.

When you include an alpha or numeric range, you must list the minimum value first in the range followed by the maximum value. For example: "0-9" and not "9-0"; "A-Z" and not "Z-A". The software uses values from the Unicode table when it chooses masked values from a range, therefore masked values are the characters that fall into the stated range based on Unicode.

Use a pipe delimiter (|) to separate multiple values. For more information about pipe-delimited lists, see [List multiple values for character and string pattern variance \[page 892\]](#).

The software counts all alphanumeric characters, spaces, and special characters when it determines the substring length. Additionally, the software outputs special characters as they are input, without masking them, when they are present in a defined substring for character pattern variance.

Character and string variance types are similar. However, there are several differences between the two variance types. For details see the topic [Differences between character and string \[page 891\]](#).

Example

The following table contains an example of the character pattern variance. For more examples see [Pattern variance examples \[page 897\]](#).

The values listed in the **Input/Possible output** column in the following table show the masked substring for the character pattern variance in bold text.

Character pattern variance

Strategy	Definition options	Input/Possible output	Notes
Mask an identification code with specific alpha or numeric values, and apply the default pattern variance to the remaining portion of the field. User sets up one <i>Definition</i> group.	<p>Undefined: Position one.</p> <p>Definition:</p> <p><i>Starting Position</i> = 2</p> <p><i>Length</i> = 2</p> <p><i>Type of Variance</i> = Character</p> <p><i>Value</i> = J-L B W-Y 2</p> <p>Undefined: Position three to the end of the string.</p>	<p>Input: 123a</p> <p>The following are possible output values:</p> <p>8KBx</p> <p>32Wt</p> <p>7LXr</p>	<p>Undefined: The software automatically masks the first position using the default pattern variance.</p> <p>Definition: The software masks position two and three using the character pattern variance and randomly chooses a character specified in the <i>Value</i> field to mask each position.</p> <p>Undefined: The software masks the fourth position to the end of the field using the default pattern variance.</p>

Increase the odds for a replacement character

To apply a little more control over the pattern variance randomization, you can increase the odds of the software choosing one value for masking over another value by including a character multiple times in the *Value* option.

For example, a value list that includes A | A | A | B causes the software to use the letter "A" about 75% more than the letter "B" when substituting characters in the defined substring.

Keep in mind that this process is random, so the actual percent may not turn out to be exactly the ratio you have set in the substring definition.

Related Information

[Pattern Variance Group \[page 879\]](#)

[Variance type: String \[page 887\]](#)

8.6.2.9.5 Variance type: String

The software masks the entire defined substring with a random character or string from the *Value* field.

The Value field can include one or more alpha or numeric characters (such as "MILK" or "2458"), spaces, special characters (such as @, #, _, &), numeric ranges, or any combination of these in a pipe-delimited list in the *Value* field.

When you include a numeric range, you must list the minimum value first in the range followed by the maximum value. For example: "0-9" and not "9-0". The software uses values from the Unicode table when it chooses masked values from a range. Therefore the software uses values from the Unicode table that fall into the stated range.

Use a pipe delimiter (|) to separate multiple values. For more information about pipe-delimited lists, see [List multiple values for character and string pattern variance \[page 892\]](#).

The software counts all alphanumeric characters, spaces, and special characters when it determines the substring length. However, the software does not retain the special characters or spaces in the output when they are present in a defined substring for string pattern variance.

Character and string variance types are similar. However, there are several differences between the two variance types. For details see the topic [Differences between character and string \[page 891\]](#).

❖ Example

The following table contains an example of the string pattern variance. For more examples see [Pattern variance examples \[page 897\]](#).

The values listed in the **Input/Possible output** column in the following table show the masked substring for the string pattern variance in bold text.

String pattern variance

Strategy	Definition options	Input/Possible output	Notes
<p>Preserve the product code, but mask the type of milk (white, chocolate, soy, and so on) with the general term MILK.</p> <p>User sets up two Definition groups.</p>	<p>Definition 1:</p> <p><i>Starting Position</i> = 1</p> <p><i>Length</i> = 5</p> <p><i>Type of Variance</i> = Preserve</p> <p><i>Value</i> = <blank></p> <p>Definition 2:</p> <p><i>Starting Position</i> = 6</p> <p><i>Length</i> = <blank></p> <p><i>Type of Variance</i> = String</p> <p><i>Value</i> = MILK</p>	<p>Input:</p> <p>5428-WTMLK</p> <p>5429-CHMLK</p> <p>5430-SOYMLK</p> <p>The following are possible output values:</p> <p>5428-MILK</p> <p>5429-MILK</p> <p>5430-MILK</p>	<p>Definition 1: The software preserves the first through the fifth positions, including the dash, as part of preserve pattern variance.</p> <ul style="list-style-type: none"> The software includes the dash when determining the substring. Additionally, the software outputs the dash as it was input because it is included in the preserve pattern variance. <p>Definition 2: The software masks position six to the end of the field with the value "MILK".</p>

Note

You could use mask out data masking for this example. However, when you use the pattern variance data masking, you can distinguish between parts of the whole string and have more control over the mask values.

Increase the odds for a replacement character

To apply a little more control over the pattern variance randomization, you can increase the odds of the software choosing one value for masking over another value by including the same string multiple times in the [Value](#) option.

For example, a value that includes VOID | VOID | VOID | ACTIVE causes the software to use the character string "VOID" about 75% more times than "ACTIVE".

Keep in mind that this process is random, so the actual percent may not turn out to be exactly the ratio you have set in the substring definition.

Related Information

[Pattern Variance Group \[page 879\]](#)

[Variance type: Character \[page 885\]](#)

8.6.2.9.5.1 Zero pad numbers

Include a zero to the left of any number in a range (to the left of the lower or higher number or to both numbers) so the mask value is left-padded with zeros for the length of the substring.

i Note

Zero-padding numbers in a range is applicable for string pattern variance only.

❖ Example

i Note

The values listed in the **Possible output** column in the following table show the masked substring for the string variance type in bold text.

Zero pad numbers in a range

Definition options	Input	Possible output	Notes
<p><i>Starting Position</i> = 1</p> <p><i>Length</i> = 5</p> <p><i>Type of Variance</i> = String</p> <p><i>Value</i> = 01-8 999</p>	04-a1099	<p>Possible output values</p> <p>00001486</p> <p>00003502</p> <p>999832</p>	<p>Definition: The software outputs the first through the fifth characters with a number from 1 up to and including 8, or the number 999.</p> <ul style="list-style-type: none"> When the software chooses a number from the range as a mask value, it zero-pads the number to the left so the masked value is the length of the defined substring (5 characters). The software does not zero-pad the number 999 because it does not contain a leading zero in the <i>Value</i> option. The software includes the dash in the input field in the position count for the substring. However the software does not output the dash as part of the string pattern variance definition. <p>Undefined: The software applies the default pattern variance to the undefined portion of the input field, character six to the end of the field, and replaces each numeric value with a random value from 0 to 9.</p>

Note

The range in the value list can also be entered as "1-08" or "01-08".

Undefined: Position six to the end of the field.

Related Information

[Variance type: String \[page 887\]](#)

[Pattern variance examples \[page 897\]](#)

8.6.2.9.6 Differences between character and string

There are several differences in software behavior between character and string pattern variance when the input *Value* field or the input substring contains specific character types.

Character versus string pattern variance

Value field	Character	String
Single alpha or numeric characters. For example: T 9 S	Allowed. The software replaces each character in a defined substring with a single alpha or numeric character that is specified in the <i>Value</i> field. For example, if the substring contains five characters, the software replaces each character with a single character from the <i>Value</i> field, for a total of five replacement characters.	Allowed. The software replaces each defined substring (the entire substring) with a value that is specified in the <i>Value</i> field. For example, if the substring contains five characters, the software replaces the five characters with a single character.
Strings of alpha or numeric characters. For example: MILK 399 abc	Not allowed. Character pattern variance accepts single alpha, numeric characters. The software issues an error if the <i>Value</i> list includes more than one character per value (except for ranges).	Allowed. The software replaces each defined substring with alpha or numeric strings that are specified in the <i>Value</i> field. For example, an input substring that consists of five characters may be replaced with a string from the <i>Value</i> field that is ten characters.
Alpha or numeric ranges. For example: D–M 2–9	Allowed. The software allows both alpha and numeric ranges. Alpha ranges can be anything from A to Z, upper or lower case. The numeric range can include single-digit numbers from 0 to 9.	Not allowed: Alpha ranges. Allowed: Numeric ranges. Numbers in ranges can have more than one digit, and they can include zero padding to the left. For example: 005–250
Spaces included with alpha characters or special characters. For example: * - a (space before asterisks, space before and after dash, space before letter “a”)	Not allowed. The software issues an error if the <i>Value</i> field includes more than one character, including spaces. i Note Single spaces and single special characters are allowed. For example, the values * a (asterisk, space, letter “a”) are allowed in the <i>Value</i> field.	Allowed. The software replaces the defined substring with a value from the <i>Value</i> field, including the spaces. i Note When the defined substring includes a space in the input, the software does not retain the space on output.

Value field	Character	String
Zero-padded individual numbers and zero-padded numbers in a range. For example: 05 010–350	Not allowed. The software issues an error if the <i>Value</i> field includes more than one character. The software allows the single-digit numbers from 0 to 9 stated individually or in a range. For example, the values 8 9 0–5 include numbers 8, 9, 0, 1, 2, 3, 4, and 5 for replacement values.	Allowed. The software allows zero-padded numbers in the <i>Value</i> field for individual numbers or numeric ranges. When the defined substring contains more characters than a zero-padded number or numeric range in the value list, the software adds zeros to the left of the number to the length of the substring. For example, a four-character substring of “1250” may be replaced with “0005” even when the listed value is “05”. Other possible masked values based on the <i>Value</i> field example at left could be “0010” or “0350”.

Related Information

Variance type: Character [\[page 885\]](#)

Variance type: String [\[page 887\]](#)

Pattern variance examples [\[page 897\]](#)

8.6.2.9.7 List multiple values for character and string pattern variance

Use pipe delimiters (|) between multiple values for character and string pattern variance masking.

You can include more than one value in the *Value* field of the *Definition* group under *Pattern Variance*. When you list more than one value, use the pipe character (|) between values. Use the back slash character (\) as an escape character to include the following characters in your value list:

- Pipe (|)
- Dash (-)
- Back slash (\)

❖ Example

The following table contains example pipe-delimited value lists that include the back slash escape character (\) before certain characters, and the dash character (-) between alpha or numeric characters for ranges. It also demonstrates how to include a space or a leading zero in the value lists.

Multiple values, escape character, zeros and spaces

Example	Description
A\C	<p>Include a dash character (-) in your value list by preceding it with a back slash (\) as an escape character.</p> <p>For the example at left, for string pattern variance, the software replaces the defined substring with "A-C" (not an inclusive range from A to C).</p>
A-C	<p>Include a dash character (-) to indicate a range. Ranges can be alpha or numeric for character pattern variance, and only numeric for string pattern variance. You do not include the back slash before the dash as an escape character for ranges.</p> <p>For the example at left, for character pattern variance, the software replaces each character in a defined substring with A, B, or C.</p>
\\ <another value>	<p>Include a back slash (\) character in your value list by preceding it with a back slash as an escape character. This applies to either character type or string type pattern variance.</p> <p>For the example at left, the software uses an escape character back slash followed by a back slash character (\\) to include the back slash character as a mask value for output.</p>
\\ <another value>	<p>Include the pipe character () in your value list by preceding it with a back slash (\) as an escape character. This applies to either character type or string type pattern variance.</p> <p>For the example at left, the software uses an escape character back slash followed by a pipe character (\\) to include the pipe character as a mask value for output.</p>

Example	Description
A 545	<p data-bbox="810 347 1374 488">For string pattern variance, include both alpha and numeric values in a value list by separating them with the pipe character () as a delimiter. For string variance, you can include values of variable length as the example at left shows.</p> <p data-bbox="810 517 1366 573">For example, a job is set up using the following definitions for a select input field:</p> <p data-bbox="810 595 932 618">Definition 1:</p> <p data-bbox="810 640 991 663"><i>Starting Position:</i> 1</p> <p data-bbox="810 674 900 696"><i>Length:</i> 5</p> <p data-bbox="810 707 1062 730"><i>Type of Variance:</i> Preserve</p> <p data-bbox="810 741 987 763"><i>Value:</i> <ignored></p> <p data-bbox="810 786 935 808">Definition 2:</p> <p data-bbox="810 831 994 853"><i>Starting Position:</i> 6</p> <p data-bbox="810 864 900 887"><i>Length:</i> 3</p> <p data-bbox="810 898 1038 920"><i>Type of Variance:</i> String</p> <p data-bbox="810 931 935 954"><i>Value:</i> A 545</p> <p data-bbox="810 987 1350 1043">For an input field of "MLK63GAL", output values may be variable length such as:</p> <ul data-bbox="820 1066 967 1133" style="list-style-type: none"> • MLK63545 • MLK63A

Example	Description
1-3 X-Z a	<p>For character pattern variance, you can include both alpha and numeric characters in a value list as the example shows at left.</p> <p>The software randomly chooses a masked value without regard to the type of character being masked. For example, the software could replace a numeric character with an alpha character from the value list.</p> <p>The job setup for the example at left includes the following Definition groups:</p> <p>Definition 1:</p> <p><i>Starting Position:</i> 1 <i>Length:</i> 5 <i>Type of Variance:</i> Preserve <i>Value:</i> <ignored></p> <p>Definition 2:</p> <p><i>Starting Position:</i> 6 <i>Length:</i> <blank> <i>Type of Variance:</i> String <i>Value:</i> 1-3 X-Z a</p> <p>Possible output values for the input field "MLK63GAL" are:</p> <ul style="list-style-type: none"> • MLK63X • MLK632
ab <another value>	<p>For string or character pattern variance, you may include a space as a value, or you could include a space before (or after) alpha characters.</p> <p>For the example at left, the software replaces a defined substring with " ab" (a space and two alpha characters), no matter how many characters the substring contains.</p>

Example	Description
07-99	<p>For string pattern variance, control the number of characters in a mask value by adding zeros to the left of a number.</p> <p>For the example at left, the user's intent is to replace the defined substring with a value that has the same number of characters as the defined <i>Length</i> of the substring. If the <i>Length</i> is 4, possible masked values could be 0008, or 0075.</p> <div data-bbox="831 651 916 680"> Note </div> <p>You only need to include one zero to the left of the first number or the second number in the range. You could include a zero before both numbers to get the same result (07-099), but it isn't necessary.</p>
A \- \\ C \\ %	<p>Include multiple characters in a pipe-delimited list by separating each value with the pipe character, and preceeding each applicable special character with a back slash as an escape character.</p> <p>For the example at left, the software uses the characters in the following list to mask a defined substring based on the character pattern variance type:</p> <ul style="list-style-type: none"> • A • - • \ • C • • % <p>The value list includes the back slash character (\) as an escape character to include the dash (-), back slash (\), and pipe () characters.</p>

8.6.2.9.8 Pattern variance examples

The following table contains examples of various pattern variance types by showing example [Definition](#) option settings, input values, and possible output values.

Strategy	Definition options	Input/Possible output	Notes
Mask the weight and the unit of measure from a product code, but preserve the product type. User sets up two Definition groups for the same input field.	Definition 1: Starting Position : 1 Length : 3 Type of Variance : Preserve Value : <ignored> Undefined : Position 4 and 5. Definition 2: Starting Position : 6 Length : 3 Type of Variance : String Value : GAL qt pt oz CUP	Input: MLK12CUP The following are possible output strings: MLK63GAL MLK18pt MLK04oz	Definition 1: Preserves the first three positions using the preserve pattern variance. Undefined: Masks the fourth and fifth positions using the default pattern variance, which replaces each numeric character with a value from 0-9. Definition 2: Masks the sixth through the eighth positions with one of the values listed using the string pattern variance. <ul style="list-style-type: none">• Notice that in some cases, the software replaces a 3-character substring with a 2-character value.

Strategy	Definition options	Input/Possible output	Notes
<p>Mask the product type and the weight from a product code, but preserve the unit of measure.</p> <p>User sets up three <i>Definition</i> groups for the same input field.</p>	<p>Definition 1:</p> <p><i>Starting Position:</i> 1 <i>Length:</i> 3 <i>Type of Variance:</i> String <i>Value:</i> ALMLK SOYMLK RCCEMLK WTMLK CHMLK</p> <p>Definition 2:</p> <p><i>Starting Position:</i> 4 <i>Length:</i> 2 <i>Type of Variance:</i> String <i>Value:</i> 01-12 32 16</p> <p>Definition 3:</p> <p><i>Starting Position:</i> 6 <i>Length:</i> <blank> <i>Type of Variance:</i> Preserve <i>Value:</i> <blank></p>	<p>Input: MLK12CUP</p> <p>The following are possible output string:</p> <p>WTMLK32CUP RCCEMLK16CUP ALMLK08CUP</p>	<p>Definition 1: Masks the first three positions using one of the values specified for string pattern variance.</p> <ul style="list-style-type: none"> The software masks the 3-character substring with values that may be longer than 3 characters. <p>Definition 2: Masks the fourth and fifth positions using the string pattern variance.</p> <ul style="list-style-type: none"> The first value listed in the <i>Value</i> field for Definition 2 is a range beginning with a zero-padded number. This ensures that the mask value is the length of the defined substring, 2 characters. <p>Definition 3: Preserves the sixth through the eighth positions.</p>

Strategy	Definition options	Input/Possible output	Notes
<p>Mask the number of paper sheets per package, and the type of packaging from the product description field.</p> <p>User sets up two <i>Definition</i> groups.</p>	<p>Definition 1:</p> <p><i>Starting Position:</i> 1 <i>Length:</i> 4 <i>Type of Variance:</i> Character <i>Value:</i> 0-9</p> <p>Undefined: Position 5.</p> <p>Definition 2:</p> <p><i>Starting Position:</i> 6 <i>Length:</i> <blank> <i>Type of Variance:</i> String <i>Value:</i> Ream Case Pack Box</p>	<p>Input: 1500/Ream</p> <p>The following are possible output strings:</p> <p>0950/Case 8945/Box 2639/Pack</p>	<p>Definition 1: Masks the first through the fourth positions with a number from 0-9.</p> <ul style="list-style-type: none"> The user could leave the first through the fifth position undefined so the software masks the substring using the default pattern variance to get similar output values. The forward slash would be output as part of the substring in this case. <p>Undefined: Outputs the forward slash (/) character in the fifth position using the default pattern variance (maintains special characters on output).</p> <p>Definition 2: Mask the sixth position to the end of the field with one of the character strings listed in the <i>Value</i> field.</p>
<p>Mask the school district, the state, and the enrollment number.</p> <p>Preserve the type of school.</p> <p>User sets up three <i>Definition</i> groups.</p>	<p>Definition 1:</p> <p><i>Starting Position:</i> 1 <i>Length:</i> 3 <i>Type of Variance:</i> String <i>Value:</i> DST</p> <p>Definition 2:</p> <p><i>Starting Position:</i> 4 <i>Length:</i> 2 <i>Type of Variance:</i> String <i>Value:</i> ST</p> <p>Undefined:</p> <p>Position 6, 7, 8, and 9.</p> <p>Definition 3:</p> <p><i>Starting Position:</i> 10 <i>Length:</i> <blank> <i>Type of Variance:</i> Preserve <i>Value:</i> <ignored></p>	<p>Input:</p> <p>INDNE7321MID BANMA7321HIGH SNBCA7321ELEM</p> <p>Possible output:</p> <p>DSTST3829MID DSTST5784HIGH DSTST0789ELEM</p>	<p>Definition 1: Masks position one to three with the string "DST".</p> <p>Definition 2: Mask the fourth and the fifth position with the string "ST".</p> <p>Undefined: Automatically masks position six through nine with the default pattern variance.</p> <p>Definition 3: Preserves the tenth position to the end of the field.</p>

i Note

The mask out variance could also mask the fields in this example. However, with pattern variance, you can distinguish between parts of the whole string and have more control over the mask values.

8.6.2.10 Number Generalization Group

Learn about the options in the [Number Generalization Group](#) and how to make settings to mask the output data with a set replacement value or a software-generated range based on your settings.

Number generalization for data masking works with all number-based input fields that you have mapped to Numeric_Data[1-6]. Map the fields in the input tab under the [Transform Input Field Name](#) column in the transform editor. Input fields can be of any data type.

If the input field contains non-numeric data with numeric-based data, the software parses the first number from the input field to use for generalization. Under certain circumstances, the software cannot represent all numbers. Read about these circumstances in “Number formats”.

Use the [Number Generalization Group](#) multiple times, but only once for each input field that you want to generalize.

Number Generalization Group, subgroups, and options

Group/Option	Description
Number Generalization Group (duplicate for each input field that you want to mask)	
Mapped Input Field	Input field to output with number generalization masking. <div><div><div>i</div>Note</div><div>The dropdown list only includes the fields that you mapped to Numeric_Data[1-6].</div></div>
Default Replacement Value	<p>Value to output when the input field value does not fall into any of the defined ranges.</p> <p>Default is Other.</p> <p>This field cannot be blank. Enter your own value or leave the default setting.</p>
Range Definition (duplicate to create multiple ranges for the same input field)	
Minimum Value	<p>Lowest value for the numeric range.</p> <p>Default is blank. If you leave this blank, the software requires that you set a value for Maximum Value.</p> <p>The following situations apply to Minimum Value:</p> <ul style="list-style-type: none">• If you enter a Minimum Value but leave the Maximum Value blank, the range is all values greater than the minimum value.• The Minimum Value cannot be greater than the Maximum Value.• The Minimum Value can be greater than or equal to -4294967296.000000 and less than or equal to 4294967295.000000.

Group/Option	Description
<i>Minimum Value Inclusive</i>	<p>Include the minimum value in the numeric range.</p> <ul style="list-style-type: none"> • Yes = Default setting. Include the value in the range. • No = Do not include the value in the range.
<i>Maximum Value</i>	<p>Highest value for the numeric range.</p> <p>Default is blank. If you leave this blank, the software requires that you set a <i>Minimum Value</i>.</p> <p>The following situations apply to <i>Maximum Value</i>:</p> <ul style="list-style-type: none"> • If you enter a <i>Maximum Value</i> but leave the <i>Minimum Value</i> blank, the range is all values less than the maximum value. • The <i>Maximum Value</i> cannot be less than the <i>Minimum Value</i>. • The value can be greater than or equal to -4294967296.000000 and less than or equal to 4294967295.000000.
<i>Maximum Value Inclusive</i>	<p>Include the maximum value in the numeric range.</p> <ul style="list-style-type: none"> • Yes = Default setting. Include the value in the range. • No = Do not include the value in the range.
<i>Replacement Value</i>	<p>Output value when the input value falls within the defined range.</p> <p>If you leave this field blank, the software generates a range based on the values that you set for the minimum and maximum options.</p>

Related Information

[Number formats \[page 863\]](#)

8.6.2.10.1 Software-generated number range examples

Use the following examples to help you understand how the software generates number ranges for number-generalized output values for Data Masking.

If you leave the *Replacement Value* in the *Range Definition* group of the *Number Generalization Group* blank, and the input value falls into the defined range, the replacement value is the software-generated number range.

i Note

For descriptions of the symbols in the range output values, see [Symbol descriptions for software-generated ranges \[page 924\]](#).

❖ Example

If you leave the *Replacement Value* blank, the software generates an output value based on the given range. For example:

- *Minimum Value* = 1
- *Minimum Value Inclusive* = Yes
- *Maximum Value* = 10
- *Maximum Value Inclusive* = No

Output value is: **[1,10)**, where “[” indicates inclusive, and “)” indicates exclusive.

❖ Example

The following table contains descriptions of possible software-generated output ranges that include mathematical symbols to indicate inclusion or exclusion of values.

Number ranges with two values

Range output	Description
[1,4]	Low and high values are inclusive: $1 \leq \text{value} \leq 4$
[1,4)	Low value inclusive. High value not inclusive: $1 \leq \text{value} < 4$
(1,4]	Low value not inclusive. High value inclusive: $1 < \text{value} \leq 4$
(1,4)	Low value and high value are not inclusive: $1 < \text{value} < 4$

❖ Example

The following table contains descriptions of possible software-generated output ranges that contain a single value or open-ended ranges.

Single-value or open-ended ranges

Range output	Description
≥ 4	All values greater than or equal to 4.
> 4	All values greater than 4.
≤ 4	All values less than or equal to 4.
< 4	All values less than 4.
4	The value 4

Related Information

[Number Generalization Group \[page 900\]](#)

[Number generalization examples \[page 903\]](#)

8.6.2.10.2 Number generalization examples

Use the following examples to help you understand how the number generalization feature works in the Data Mask transform.

❖ Example

You have a database that includes age data. You want to group the records into the following groups:

- First range is for ages 12 or younger. Replace with *Default Replacement Value* "Other".
- Second range is for ages 13 to 19. Replace with "Teenager".
- Third range is for ages 20 to 24. Replace with "Young Adult".
- Fourth range is for ages 30 and older. Replace with "Adult".

Job set up:

- *Mapped Input Field* = Age
- *Default Replacement Value* = Other
- *Range Definition* (1)
 - Minimum Value* =
 - Minimum Value Inclusive* = YES
 - Maximum Value* = 12
 - Maximum Value Inclusive* = YES
 - Replacement Value* =
- *Range Definition* (2)
 - Minimum Value* = 13
 - Minimum Value Inclusive* = YES
 - Maximum Value* = 19
 - Maximum Value Inclusive* = YES
 - Replacement Value* = Teenager
- *Range Definition* (3)
 - Minimum Value* = 19
 - Minimum Value Inclusive* = NO
 - Maximum Value* = 25
 - Maximum Value Inclusive* = NO
 - Replacement Value* = Young Adult
- *Range Definition* (4)
 - Minimum Value* = 30
 - Minimum Value Inclusive* = YES
 - Maximum Value* =
 - Maximum Value Inclusive* = YES
 - Replacement Value* = Adult

Sample data

Input	Output	Explanation
4	<=12	Falls into the first range, which has no minimum. The <i>Replacement Value</i> is blank so the software outputs the software-generated range.
12	<=12	Falls into the first range, which has no minimum. The maximum value is inclusive. The <i>Replacement Value</i> is blank so the software outputs the software-generated range.
13	Teenager	Falls into the second range because it is equal to the minimum value of 13. Minimum value is inclusive.
17	Teenager	Falls into the second range because it is greater than 13 but less than 19.
19	Teenager	Falls into the second range because it is equal to the maximum of 19. Maximum value is inclusive.
25	Other	The third range specifies 25 as the <i>Maximum</i> , but the <i>Maximum Number Inclusive</i> is set to <i>NO</i> . Therefore, 25 does not fall into any range so the software outputs the <i>Default Replacement Value</i> .
32	Adult	Falls into the fourth range because it is greater than 30.
\$13.75	Teenager	Falls into the second range. The software parses the field until it finds the first number, which is 13.75. The number 13.75 is within the range of 13 - 19.
Hi mom	NULL	The software cannot parse number-based data from the input data. Therefore the output is NULL.
NULL	NULL	The software cannot parse number-based data from the input data. Therefore the output is NULL.

Related Information

Number Generalization Group [page 900]
Software-generated number range examples [page 901]

8.6.2.11 Date Generalization Group

Use the *Date Generalization Group* on date-based input fields to mask the output data with a common label or a date range.

Date generalization for data masking works with all date-based input fields that you have mapped to Date[1-6]. Map the fields in the input tab under the *Transform Input Field Name* column in the transform editor. Input fields can be of any data type.

Use the *Date Generalization Group* once for each input field that you want to generalize.

There are two types of ranges that you can define:

- *Range Definition*: You define the date ranges. Use the *Range Definition* group multiple times on an eligible input field.
- *Auto Range Definition*: The software calculates date ranges based on your settings. Use the *Auto Range Definition* group once per eligible input field.

i Note
You may not use both types of ranges on the same input field.

Date Generalization Group, subgroups, and options

Group/Option	Description
<i>Date Generalization Group</i> (duplicate for each input field that you want to mask)	
<i>Mapped Input Field</i>	Input field to output with date generalization masking. <div>i Note The dropdown list only includes the fields that you mapped to Date[1-6].</div>
<i>Default Replacement Value</i>	Value to output when the input field value does not fall into any of the defined ranges. The default is <i>Other</i> . This field cannot be blank. Enter your own value or leave the default setting.
<i>Range Definition</i> (duplicate to set multiple ranges for the same input field)	

Group/Option	Description
<i>Minimum Date</i>	<p>Earliest date for the range.</p> <p>Default is blank. If you leave the option blank, the software requires that you enter a date for the <i>Maximum Date</i> option.</p> <p>The following situations apply to <i>Minimum Date</i>:</p> <ul style="list-style-type: none"> • The <i>Minimum Date</i> cannot be later than or equal to the <i>Maximum Date</i>. • If you specify a <i>Minimum Date</i> but you do not specify a <i>Maximum Date</i>, the software uses a date range of later than or equal to the <i>Minimum Date</i>. • The <i>Minimum Date</i> cannot be earlier than January 1, 1900.
<i>Minimum Date Inclusive</i>	<p>Specifies to include the value for <i>Minimum Date</i> in the date range.</p> <ul style="list-style-type: none"> • <i>Yes</i> = Default setting. Include in the range. • <i>No</i> = Do not include in the range.
<i>Maximum Date</i>	<p>Maximum date for the range.</p> <p>Default is blank. If you leave the option blank, the software requires that you enter a date for the <i>Minimum Date</i> option.</p> <p>The following situations apply to <i>Maximum Date</i>:</p> <ul style="list-style-type: none"> • The <i>Maximum Date</i> cannot be earlier than or equal to the <i>Minimum Date</i>. • If you specify a <i>Maximum Date</i> but you do not specify a <i>Minimum Date</i>, the software uses a range of earlier than or equal to the <i>Maximum Date</i>. • The <i>Maximum Date</i> cannot be later than December 31, 2099.
<i>Maximum Date Inclusive</i>	<p>Specifies to include the value for <i>Maximum Date</i> in the date range.</p> <ul style="list-style-type: none"> • <i>Yes</i> = Default setting. Include in the range. • <i>No</i> = Do not include in the range.
<i>Replacement Value</i>	<p>Specifies the output value when the input value falls within the defined range in the <i>Range Definition</i> options.</p> <p>Default is blank.</p> <p>When you leave the <i>Replacement Value</i> blank, the software outputs a system-generated range based on your settings.</p>
<i>Auto Range Definition</i> (Use once per applicable input field. If you have defined the Range Definition group for the same input field, you may not use this group.)	

Group/Option	Description
<i>Auto Range Scale</i>	<p>Defines the scale on which to base the auto range.</p> <ul style="list-style-type: none"> • <i>Not_In_Use</i> = Default setting. Indicates that you are not using auto range for the specified input field. This setting is appropriate when you complete the <i>Range Definition</i> options for the input field, or when you do not use this feature. • <i>Calendar_Year</i> = Group records based on the calendar year. The software defines a calendar year as 1/1/yyyy to 12/31/yyyy. • <i>Calendar_Month</i> = Group records based on the calendar month. The software defines a calendar month as mm/01/yyyy to mm/eom/yyyy, where “eom” is end of month. <div> <p>i Note</p> <p>For both <i>Calendar_Year</i> and <i>Calendar_Month</i>, the software can use settings that you make in the <i>Start Date (Inclusive)</i> and <i>End Date (Inclusive)</i> as part of the range.</p> </div> <p>If you choose a setting that is not compatible with the <i>Date Format</i> in the <i>Auto Range Scale</i> option, the software issues an error. For example, the software issues an error when <i>Auto Range Scale</i> is set to <i>Calendar_Year</i> and <i>Date Format</i> is set to <i>Month</i>.</p>
<i>Duration</i>	<p>Number of years or months to include in the range.</p> <p>Default is <i>1</i>.</p> <p>If you set the <i>Duration</i> to greater than 1, you must also set a value for <i>Start Date (Inclusive)</i>. If you do not set a start date and the duration is set to greater than 1, the software issues an error.</p>

Group/Option	Description
<i>Start Date (Inclusive)</i>	<p>Starting date in auto range.</p> <p>Default is blank.</p> <p>The following situations apply to <i>Start Date (Inclusive)</i>:</p> <ul style="list-style-type: none"> • If you enter a <i>Start Date (Inclusive)</i> and an <i>End Date (Inclusive)</i>, the start date must be earlier than the end date. • If the <i>Duration</i> is set to 1, you can leave the <i>Start Date (Inclusive)</i> and <i>End Date (Inclusive)</i> blank. • The minimum start date cannot be earlier than January 1, 1900. <p>The software outputs the <i>Default Replacement Value</i> for all dates that fall outside of the range.</p>
<i>End Date (Inclusive)</i>	<p>Ending date in auto range.</p> <p>Optional. Default is blank.</p> <p>The software uses this date as the last date in the calculated range.</p> <p>The following situations apply to <i>End Date (Inclusive)</i>:</p> <ul style="list-style-type: none"> • If you enter both a <i>Start Date (Inclusive)</i> and an <i>End Date (Inclusive)</i>, the end date must be later than the start date. • If the <i>Duration</i> is set to 1, you can leave the <i>Start Date (Inclusive)</i> and <i>End Date (Inclusive)</i> blank. • The maximum end date cannot be later than December 31, 2099. <p>The software outputs the <i>Default Replacement Value</i> for all dates that fall outside of the range.</p>
<i>Auto Range Replacement Value</i> (specific to the auto range defined in the parent group <i>Auto Range Definition</i>)	

Group/Option	Description
<i>Date Format</i>	<p>Determines the format of the output <i>Auto Range Replacement Value</i>.</p> <ul style="list-style-type: none"> • <i>Month_Day_Year</i> Default setting • <i>Day_Month_Year</i> • <i>Year_Day_Month</i> • <i>Year_Month_Day</i> • <i>Year</i> • <i>Year_Month</i> • <i>Month_Year</i> • <i>Month</i> <div> <p>i Note</p> <p>If the setting that you choose is not compatible with the setting in <i>Auto Range Scale</i>, the software issues an error.</p> </div>
<i>Date Delimiter</i>	<p>Determines the delimiter to use in the <i>Auto Range Replacement Value</i>.</p> <ul style="list-style-type: none"> • <i>Slash</i> Default setting • <i>Dash</i> • <i>Backslash</i> • <i>Space</i> • <i>None</i> • <i>Period</i> • <i>Chinese_Japanese</i>
<i>Numeric Format</i>	<p>Determines the numeric format to use in the <i>Auto Range Replacement Value</i>.</p> <ul style="list-style-type: none"> • <i>Arabic_Numbers</i> Default setting • <i>Chinese_Japanese_Numbers</i>
<i>Enable Zero Pad</i>	<p>Pad a one-digit number with zero when the format includes the month and day.</p> <ul style="list-style-type: none"> • <i>No</i> = Default setting. Do not pad dates with zeros. Example: Jan 2, 2016 is output as Jan 2 2016. • <i>Yes</i> = Pad dates with zeros. Example: Jan 2, 2016 is output as Jan 02 2016.

Group/Option	Description
<i>Month Format</i>	<p>Determines the month format to use in the <i>Auto Range Replacement Value</i>.</p> <ul style="list-style-type: none"> • <i>Numeric</i> = Default setting. Represent the month with a number. • <i>Short_Text</i> = Represent the month using the month abbreviation. • <i>Full_Text</i> = Represent the month using the full spelling.
<i>Language</i>	<p>Determines the language to use in the <i>Auto Range Replacement Value</i>. This setting is applicable when the <i>Month Format</i> is set to <i>Short_Text</i> or <i>Full_Text</i>.</p> <p><i>English</i> Default setting</p> <div> <p>i Note</p> <p>The software does not verify that the user-defined default language corresponds to the language of the input month.</p> </div>
<i>Year Format</i>	<p>Specifies the number of digits to use for the year.</p> <ul style="list-style-type: none"> • <i>Full_Year</i> = Default setting. Always output a 4-digit year. • <i>Short_Year</i> = Always output a 2-digit year.

Related Information

[Examples range definition \[page 912\]](#)

[Examples auto defined ranges \[page 916\]](#)

8.6.2.11.1 User-defined ranges and replacement values

Set date ranges in the *Range Definition* group in the *Date Generalization Group* to identify specific input date-based data and to generalize the output with a set value.

Use the *Range Definition* options multiple times for a single input field to set multiple ranges.

i Note

If you set multiple ranges for the same input field, make sure that the ranges do not overlap.

i Note

When you use *Range Definition* for a specific input field, you cannot also use *Auto Range Definition* for the same input field.

Enter a value for *Replacement Value* for each range that you set, or accept the default of blank.

- When you set a value and an input date meets the range parameters, the software outputs the value that you set in *Replacement Value*.
- When you accept the default of blank and an input date meets the range parameters, the software generates a date range to output based on the minimum and maximum or start and end dates defined in this range definition.

How system-generated ranges work

For calendar year ranges, the software starts the first range at the *Start Date (inclusive)* and ends the range on December 31, <year>. The software calculates the range based on the set *Duration*. Each subsequent range starts on January 1 and ends on December 31 of the calculated year. The last range ends on the set *End Date (inclusive)*.

For calendar month ranges, the software starts the first range at the *Start Date (inclusive)* and ends the range on the last day of the appropriate month and calculated year. The software calculates the range based on the set *Duration*. Each subsequent range starts on the first day of the month and ends on the last day of the month of the calculated year. The last range ends on the date specified in the *End Date (inclusive)*.

How the generated ranges are output

The software outputs a generated range using mathematical symbols that indicate less than, equal to, inclusive, or exclusive. The following are examples of output generated ranges.

❖ Example

Date ranges with two values:

Range output	Description
[3/17/2017, 7/4/2017]	Both low and high dates are inclusive: March 17, 2017 <= date <= July 4, 2017
[3/17/2017, 7/4/2017)	Low date inclusive. High date not inclusive: March 17, 2017 <= date < July 4, 2017
(3/17/2017, 7/4/2017]	Low date not inclusive. High date inclusive: March 17, 2017 < date <= July 4, 2017

Range output	Description
(3/17/2017, 7/4/2017)	Low and high dates are not inclusive: March 17, 2017 < date < July 4, 2017

❖ Example

Single-value or open-ended ranges

Range output	Description
>= 3/17/2017	All dates later than or equal to March 17, 2017.
> 3/17/2017	All dates later than March 17, 2017.
<= 3/17/2017	All dates earlier than or equal to March 17, 2017.
< 3/17/2017	All dates earlier than March 17, 2017.
3/17/2017	The date is March 17, 2017

Related Information

[Symbol descriptions for software-generated ranges \[page 924\]](#)

[Date Generalization Group \[page 905\]](#)

8.6.2.11.1 Examples range definition

Review the examples to help you understand how to set date ranges in the *Range Definition* options in the *Date Generalization Group*.

❖ Example

You want to customize an advertising campaign based on generation groups. You create three databases, each containing records that qualify for the generational groups based on birth date. The generational groups are:

- Baby Boomer
- Generation X
- Millennial

Date Generalization Group, *Range Definition* option settings:

Date Generalization Group

- *Date Generalization*
Mapped Input Field = BirthDate
Default Replacement Value = Other
- *Range Definition (1)*
Minimum Date =
Minimum Date Inclusive = YES
Maximum Date = January 1, 1945
Maximum Date Inclusive = NO
Replacement Value = N/A
- *Range Definition (2)*
Minimum Date = Jan 1, 1945
Minimum Date Inclusive = YES
Maximum Date = 12/31/1964
Maximum Date Inclusive = YES
Replacement Value = Baby Boomer
- *Range Definition (3)*
Minimum Date = Jan 1, 1965
Minimum Date Inclusive = YES
Maximum Date = 12/31/1976
Maximum Date Inclusive = YES
Replacement Value = GEN-X
- *Range Definition (4)*
Minimum Date = 01/01/1977
Minimum Date Inclusive = YES
Maximum Date = December 31, 1995
Maximum Date Inclusive = YES
Replacement Value = Millennial

Sample data

Input	Output	Explanation
7/31/1924	N/A	The input date falls into the first range definition of less than January 1, 1945. The software outputs the replacement value "N/A"
1/1/45	Baby Boomer	The input date falls into the second range definition. 1/1/45 is greater than or equal to 1/1/1945 and less than or equal to 12/31/1964.

Input	Output	Explanation
Nov 4, 1974	GEN-X	The input date falls into the third range definition. Nov 4, 1974 is greater than or equal to 1/1/1965 and less than or equal to 12/31/1976.
July 4, 1988	Millennial	The input date falls into the fourth date range definition. July 4, 1988 is greater than or equal to 1/1/1977 and less than or equal to 12/31/1995.
December 6, 2016	Other	The input date does not fall into any defined range. December 6, 2016 is greater than the maximum date of 12/31/1995. Therefore, the software outputs the <i>Default Replacement Value</i> "Other".
Born: 11/4/1960	Baby Boomer	The input data contains both text and date-based data. The software parses the date data. The input date falls into the second date range definition. 11/4/1960 is greater than or equal to 1/1/1945 and less than or equal to 12/31/1964.
NULL	NULL	The input does not contain any parsable date data. The software issues a warning.
Hi Mom!	NULL	The input does not contain any parsable date data. The software issues a warning.

Related Information

[Date Generalization Group \[page 905\]](#)

8.6.2.11.2 Auto ranges and replacement values

Use options in the [Auto Range Definition](#) group to have the software calculate date ranges to use for Date Generalization. Use the [Auto Range Definition](#) options once per input field.

i Note

When you use [Auto Range Definition](#) for a specific input field, you cannot also use [Range Definition](#) for the same input field.

Unlike the multiple ranges you can set for an input field in the user-define date range options, you enter a start and end date and the software calculates all of the ranges.

Instead of setting a value for an output replacement, the software calculates ranges to output. You define the format of the generated ranges by making settings in the [Auto Range Replacement Value](#) group.

How the software calculates ranges

- The software calculates ranges based on calendar year in the following manner:
 - The first range starts at the set [Start Date \(Inclusive\)](#) and goes to December 31 of the year specified by the [Duration](#) setting.
 - All subsequent ranges start on January 1 and end on December 31 of the calculated year.
 - The last range starts on January 1 of the calculated year and ends on the specified [End Date \(Inclusive\)](#) setting.
- The software calculates ranges based on calendar month in the following manner:
 - The first range starts at the set [Start Date \(Inclusive\)](#) and goes to the last day of the stated month based on the [Duration](#) setting.
 - All subsequent ranges start on the first of the calculated month and end on the last day of the calculated month.
 - The last range starts on the first of the calculated month and ends on the specified [End Date \(Inclusive\)](#) setting, if the user has set an [End Date \(Inclusive\)](#) date.

Related Information

[Date Generalization Group \[page 905\]](#)

[Examples auto defined ranges \[page 916\]](#)

[Calculated partial ranges for Auto Range Definition \[page 917\]](#)

8.6.2.11.2.1 Examples auto defined ranges

Use these examples to understand how the software calculates auto-defined ranges based on your settings in the [Date Generalization Group](#).

❖ Example

You want to run date generalization on the `<PublishDate>` field in your database. You want to group the dates in five-year durations beginning with January 1, 2000. If any date is earlier than the start date of January 1, 2000, you want the software to output "Not this millennium" in the `<PublishDate>` field.

- [Date Generalization Group](#)

[Mapped Input Field](#) = PublishDate

[Default Replacement Value](#) = Not this millennium

- [Range Definition](#)

You do not enter values for this group because the software does not allow both range definition and auto range definition for the same input field.

[Minimum Date](#) =

[Minimum Date Inclusive](#) = YES

[Maximum Date](#) =

[Maximum Date Inclusive](#) = YES

[Replacement Value](#) =

- [Auto Range Definition](#)

[Auto Range Scale](#) = CALENDAR_YEAR

[Duration](#) = 5

[Start Date \(Inclusive\)](#) = January 1, 2000

[End Date \(Inclusive\)](#) =

- [Auto Range Replacement Value](#)

[Date Format](#) = YEAR_MONTH

[Date Delimiter](#) = SLASH

[Numeric Format](#) = ARABIC_NUMBERS

[Enable Zero Pad](#) = YES

[Month Format](#) = NUMERIC

[Language](#) = ENGLISH

[Year Format](#) = FULL_YEAR

Sample data

Input	Output	Explanation
7/31/1924	Not this millennium	The input date is out of range because it is earlier than the start date of January 1, 2000. The software outputs the Default Replacement Value .

Input	Output	Explanation
1/1/2000	[2000/01, 2004/12]	The input date is in range because it is equal to the Start Date (Inclusive) value. The output is in YEAR_MONTH date format as specified in the options.
Dec 6, 2016	[2015/01, 2019/12]	The input date is in range because it is later than the Start Date (Inclusive) and there is no End Date (Inclusive) .
Date: 2016	NULL	The input does not contain any parsable date data. The software issues a warning that the date could not be parsed.
December 10	NULL	The input does not contain any parsable date data. The software issues a warning that the date could not be parsed.

Related Information

[Date Generalization Group \[page 905\]](#)

8.6.2.11.2.2 Calculated partial ranges for Auto Range Definition

For calendar year and calendar month ranges, the software may use a partial range for the first or last range of the duration based on the set start and end dates that you enter in the job set up.

SAP Data Services calculates auto-ranges based on your settings in the [Auto Range Definition](#) group for either a calendar month or a calendar year. Entering a start date for the first of a year and an end date for the last day of the year enables the software to calculate full year ranges. However, enter a start date that doesn't start on 1/1/yyyy causes the first year range to be a partial year. Enter an end date that doesn't end on 12/31/yyyy causes the last year range to be a partial year. The same reasoning is true for month calculations. Enter the start date that is not mm/1/yyyy and the first range is a partial month. Enter the end date that is not mm/31/yyyy and the last range is partial month.

The software does not require that you enter the start date and end date so that the software calculates full durations.

i Note

If you do not enter a start or end date, the software requires that you set the *Duration* to 1.

❖ Example

For *Auto Range Scale* of *CALENDAR_YEAR* with *Duration* of 1, the start dates result in the calculated ranges shown in the following table:

Full year	Partial year
Start Date (inclusive) = 1/1/2017 End Date (inclusive) = 12/31/2020	Start Date (inclusive) = 7/4/2017 End Date (inclusive) = 12/31/2020
First calculated range = 1/1/2017 to 12/31/2017	First calculated range = 7/4/2017 to 12/31/2017
Second calculated range = 1/1/2018 to 12/31/2018	Second calculated range = 1/1/2018 to 12/31/2018
Third calculated range = 1/1/2019 to 12/31/2019	Third calculated range = 1/1/2019 to 12/31/2019
Fourth calculated range = 1/1/2020 to 12/31/2020	Fourth calculated range = 1/1/2020 to 12/31/2020

To further illustrate full and partial year and month range calculations, see the examples for both calendar year and calendar month.

Related Information

[Auto ranges and replacement values \[page 915\]](#)

8.6.2.11.2.2.1 Examples of software-calculated ranges based on calendar year

Use these examples to help you understand how the software calculates and outputs date ranges based on calendar year and date format.

❖ Example

- *Auto Range Definition*
Auto Range Scale = CALENDAR_YEAR
Duration = 1
Start Date (Inclusive) =
End Date (Inclusive) =
- *Auto Range Replacement Value*
Date Format = <see table>
Date Delimiter = SLASH

Numeric Format = ARABIC_NUMBERS
Enable Zero Pad = YES
Month Format = NUMERIC
Language = ENGLISH
Year Format = FULL_YEAR

The software calculates the ranges based on the input date because there are no start and end dates specified. The input date is July 4, 2016. The output values differ based on the set *Date Format* in the first column. All dates in the Output column are inclusive so the ranges are enclosed with square brackets.

Note that not all Date Format values are shown in the examples

Input date = July 4, 2016		
Date Format =	Output	Notes
YEAR_DAY_MONTH	[2016/01/01, 2016/31/12]	The output appears in yyyy/dd/mm format.
YEAR_MONTH_DAY	[2016/01/01, 2016/12/31]	The output appears in yyyy/mm/dd format.
YEAR	2016	The output is not a range. Because the scale is calendar year, and the duration is 1, the only possible output is the year. The format is yyyy.
YEAR_MONTH	[2016/01, 2016/12]	The output appears in yyyy/mm format.
MONTH	NULL	No output. This date format causes the software to issue an error. The <i>Auto Range Scale</i> of Calendar Year and the <i>Date Format</i> are not compatible.

❖ Example

- *Auto Range Definition*

Auto Range Scale = CALENDAR_YEAR
Duration = 2
Start Date (Inclusive) = July 15, 2013
End Date (Inclusive) =

- *Auto Range Replacement Value*

Date Format = <see table>
Date Delimiter = DASH
Numeric Format = ARABIC_NUMBERS
Enable Zero Pad = NO
Month Format = FULL_TEXT

Language = ENGLISH
Year Format = FULL_YEAR

The start date begins in the middle of a year. Therefore, the first software-calculated 2-year range isn't a full two years. The first two software-calculated date ranges are as follows:

- First 2-year range: July 15, 2013 to December 31, 2014
- Second 2-year range: January 1, 2015 to December 31, 2016

In the following table, the input date is July 4, 2016. This date falls into the second calculated range. The output values differ based on the set *Date Format* in the first column. All dates in the Output column are inclusive so the ranges are enclosed with square brackets.

Note that not all Date Format values are shown in the examples

Input date = July 4, 2016		
Date Format	Output	Notes
DAY_MONTH_YEAR	[1-January-2015, 31-December-2016]	The output appears with dashes and full text month format: d-full text month-yyyy.
YEAR	[2015, 2016]	The output appears with dashes and includes just the year format: yyyy.
YEAR_DAY_MONTH	[2015-1-January, 2016-31-December]	The output appears with dashes and full text month format: yyyy-d-full text month.
MONTH_DAY_YEAR	[January-1-2015, December-31-2016]	The output appears with dashes and full text month format: full text month-d-yyyy.

❖ Example

- *Auto Range Definition*

Auto Range Scale = CALENDAR_YEAR

Duration = 5

Start Date (Inclusive) = July 15, 2013

End Date (Inclusive) =

- *Auto Range Replacement Value*

Date Format = <see table>

Date Delimiter = PERIOD

Numeric Format = ARABIC_NUMBERS

Enable Zero Pad = YES

Month Format = NUMERIC

Language = ENGLISH

Year Format = FULL_YEAR

The input date is November 22, 2016. This date falls into the first 5-year calculated range:

- First 5-year range: 7/15/2013 to 12/31/2017

Note that not all Date Format values are shown in the examples

Input date = Nov 22, 2013		
Date Format	Output	Notes
DAY_MONTH_YEAR	[15.07.2013, 31.12.2017]	The output appears in dd.mm.yyyy format.
YEAR_DAY_MONTH	[2013.15.07, 2017.31.12]	The output appears in yyyy.dd.mm format.
YEAR	2013, 2017	The output appears in yyyy format.
YEAR_MONTH	[2013.07, 2017.12]	The output appears in yyyy.mm format.
DAY_YEAR_MONTH	[15.2013.07, 31.2017.12]	The output appears in dd.yyyy.mm format.

Related Information

[Date Generalization Group \[page 905\]](#)

8.6.2.11.2.2 Examples of software-calculated ranges based on calendar month

Use these examples to help you understand how the software calculates and outputs date ranges based on calendar month and date format.

❖ Example

- *Auto Range Definition*

Auto Range Scale = CALENDAR_MONTH

Duration = 1

Start Date (Inclusive) =

End Date (Inclusive) =

- *Auto Range Replacement Value*

Date Format = <see table>

Date Delimiter = SLASH

Numeric Format = ARABIC_NUMBERS

Enable Zero Pad = YES

Month Format = NUMERIC

Language = ENGLISH

Year Format = FULL_YEAR

The software uses the input date to calculate ranges because there are no start or end dates specified. The input date is July 4, 2016. All ranges in the output column appear in square brackets because the start and end dates are inclusive. The output values are the same, however, they change based on the listed date format in the first column.

Note that not all date formats are shown in the examples.

Input date = July 4, 2016		
Date Format =	Output	Notes
DAY_MONTH_YEAR	[01/07/2016, 31/07/2016]	The output appears in dd/mm/yyyy format.
MONTH_DAY_YEAR	[07/01/2016, 07/31/2016]	The output appears in mm/dd/yyyy format.
YEAR_DAY_MONTH	[2016/01/07, 2016/31/07]	The output appears in yyyy/dd/mm format.
YEAR	NULL	The software issues an error because the <i>Auto Range Scale</i> of calendar month and the <i>Date Format</i> are incompatible.
MONTH_YEAR	07/2016	The output appears in mm/yyyy format. The output is not a range. The software cannot calculate a range based on the scale, date format, and duration.
MONTH	07	The output appears in mm format. The output is not a range. The software cannot calculate a range because of the scale, date format, and duration.

❖ Example

- *Auto Range Definition*

Auto Range Scale = CALENDAR_MONTH

Duration = 3

Start Date (inclusive) = July 15, 2013

End Date (inclusive) =

- *Auto Range Replacement Value*

Date Format = <see table>
Date Delimiter = PERIOD
Numeric Format = ARABIC_NUMBERS
Enable Zero Pad = YES
Month Format = NUMERIC
Language = ENGLISH
Year Format = FULL_YEAR

The first software-calculated range begin with a partial month:

- First software-calculated range: 7/15/2013 to 9/30/2013
- Second software-calculated range: 10/1/2013 to 12/31/2013

The input date is July 15, 2013. It falls into the first software-calculated range. The start date and end date are inclusive, so the output ranges appear in square brackets.

Note that not all date formats are shown in the examples.

Input date = July 15, 2016		
Date Format =	Output	Notes
MONTH_DAY_YEAR	[07.01.2016, 09.30.2016]	The output appears in mm.dd.yyyy format.
YEAR_MONTH_DAY	[2016.07.01, 2016.09.30]	The output appears in yyyy.mm.dd format.
YEAR	NULL	The software issues an error because the <i>Auto Range Scale</i> of calendar month and the <i>Date Format</i> of year are incompatible.
YEAR_MONTH	[2016.07, 2016.09]	The output appears in yyyy.mm format.
MONTH_YEAR	[07.2016, 09.2016]	The output appears in mm.yyyy format.
MONTH	[07, 09]	The output appears in mm format.

❖ Example

- *Auto Range Definition*

Auto Range Scale = CALENDAR_MONTH
Duration = 3
Start Date (inclusive) = July 15, 2013
End Date (inclusive) =

- *Auto Range Replacement Value*

Date Format = <see table>
Date Delimiter = DASH

Numeric Format = ARABIC_NUMBERS
Enable Zero Pad = NO
Month Format = SHORT_TEXT
Language = ENGLISH
Year Format = SHORT_YEAR

The input date is July 21, 2013. It falls within the first software-calculated range of 7/15/2013 to 9/30/2013. Therefore, the first date of the range does not fall at the beginning of the month. The software does not issue a warning.

Input date = July 21, 2013		
Date Format =	Output	Notes
DAY_MONTH_YEAR	[15-Jul-13, 30-Sept-13]	The output appears in dd-short text month-yy format.
MONTH_DAY_YEAR	[Jul-15-13, Sept-30-13]	The output appears in short text month-dd-yy format.
YEAR_DAY_MONTH	[13-15-Jul, 13-30-Sept]	The output appears in yy-dd-short text month format.
YEAR	NULL	The software issues an error. The settings in <i>Auto Range Scale</i> and the <i>Date Format</i> are incompatible.
YEAR_MONTH	[13-Jul, 13-Sept]	The output appears in yy-short text month format.
MONTH	[Jul, Sept]	The output appears in short text month format.

Related Information

[Date Generalization Group \[page 905\]](#)

8.6.2.11.3 Symbol descriptions for software-generated ranges

The mathematical equation symbols that the software uses in generated output values indicate if a value is included or not included in the range.

The following table describes the symbols that the software uses in the software-generated ranges for both number generalization and date generalization.

Symbol	Description	Meaning
[Left square bracket	Low value is inclusive
]	Right square bracket	High value is inclusive
(Left parenthesis	Low value is exclusive
)	Right parenthesis	High value is exclusive
>= "val"	Right pointing arrow and equal sign	Value is greater than or equal to "val".
> "val"	Right pointing arrow	Value is greater than "val".
<= "val"	Left pointing arrow and equal sign	Value is less than or equal to "val".
< "val"	Left pointing arrow	Value is less than "val".
"val"	Exact value	An exact value has no symbols

Related Information

[Number Generalization Group \[page 900\]](#)

[Software-generated number range examples \[page 901\]](#)

[Date Generalization Group \[page 905\]](#)

8.6.3 DQM Microservices

With the DQM Microservices transform, you can configure and execute SAP Data Quality Management, microservices for location data services within SAP Data Services.



SAP Data Quality Management, microservices for location data offers cloud-based microservices for address cleansing, geocoding, and reverse geocoding. You can embed address cleansing and enrichment services within any business process or application so that you can quickly reap the value of complete and accurate address data.

The DQM Microservices transform supports the following DQM microservices:

- ***addressCleanse service***: Address cleansing corrects, parses, standardizes, validates, and enhances address data, and assigns geocodes.
- ***reverseGeo service***: Reverse geocoding translates latitude and longitude coordinates into address(es).

For more information about DQM microservices, see the documentation set at <https://help.sap.com/viewer/d95546360fea44988eb614718ff7e959/Cloud/en-US>.

i Note

The DQM Microservices functionality is available through a productive landscape and a trial landscape.

For more information about configuring an SAP DQM Microservices datastore and using SAP Data Quality Management, microservices for location data within Data Services, see “Connecting to SAP Data Quality Management, microservices for location data” in the *Supplement for SAP*.

Parent topic: [Platform transforms \[page 833\]](#)

Related Information

[Case \[page 833\]](#)

[Data Mask \[page 836\]](#)

[Map_Operation \[page 938\]](#)

[Merge \[page 942\]](#)

[Query \[page 943\]](#)

[Row_Generation \[page 970\]](#)

[SQL \[page 972\]](#)

[User-Defined \[page 975\]](#)

[Validation \[page 983\]](#)

[XML_Map \[page 994\]](#)

8.6.3.1 Content objects

We provide content objects to help you create and complete transforms.

Transform configurations

A transform configuration is a transform with preconfigured input fields, output fields, and options that can be used in multiple data flows. These are useful if you repeatedly use a transform with specific options and input and output fields.

When Data Services is installed, read-only transform configurations are provided for the Data Quality transforms.

You can use transform configurations in your data flows or as an example of a typical transform. After you place an instance of the transform configuration in a data flow, you can override these preset defaults. You can also create your own transform configuration, either by replicating an existing transform configuration or creating a new one.

Sample blueprints and other objects

We have created Data Quality blueprints and other content objects to help you set up Data Services jobs. We've identified a number of common scenarios that you are likely to perform with Data Services. For each scenario, we've included a blueprint that is already set up to solve the business problem in that scenario.

Related Information

[Downloading blueprints and other content objects \[page 444\]](#)

[Transform configurations \[page 440\]](#)

8.6.3.2 DQM Microservices options

The Options group includes settings that control how the DQM Microservices transform connects to the datastore and specifies the services and configurations used for SAP Data Quality Management, microservices for location data.

Option	Description
Datastore	<p>Specifies an existing SAP DQM Microservices datastore, which configures connection information that Data Services uses to access DQM microservices.</p> <p>To add a datastore, create a new SAP DQM Microservices datastore using the Datastore Editor. For detailed instructions, see the <i>Supplement for SAP</i>.</p>
Service	<p>Specifies the DQM microservices service to use.</p> <p><i>addressCleanse</i>: corrects, parses, standardizes, validates, and enhances address data, and assigns geocodes.</p> <p><i>reverseGeo</i>: translates latitude and longitude coordinates into address(es).</p> <p>This option becomes available after you specify a datastore.</p>

Option	Description
Configuration	<p>Contains the available configurations for the selected service in the default configuration of the specified datastore. In DQM microservices, you can define configurations for the addressCleanse service. The configurations tell the addressCleanse service how to map or route addresses for processing. (For example, French addresses are routed for French-specific processing, German addresses are routed for German-specific processing, and so on.) The configurations also contain the addressCleanse service options.</p> <p>If the Service option is set to reverseGeo, None is the only available value.</p> <p>If the Service option is set to addressCleanse, all available DQM microservices configurations for the specified datastore are displayed.</p> <p>This option becomes available after you specify a service.</p> <p>For more information about creating configurations, see the DQM microservices documentation at https://help.sap.com/viewer/d95546360fea44988eb614718ff7e959/Cloud/en-US.</p>
Settings	<p>Open the Settings window, which displays the settings retrieved from DQM microservices. The options displayed are different for each type of supported service.</p> <p>For descriptions of the available settings for each service, see the DQM microservices documentation at https://help.sap.com/viewer/d95546360fea44988eb614718ff7e959/Cloud/en-US.</p> <p>This option becomes available if the Configuration option is set to None. The option is not available if a configuration has been selected.</p>

8.6.3.3 Input fields for the DQM Microservices transform

The available DQM Microservices transform input fields are determined by the fields available in the service or configuration that has been selected for SAP Data Quality Management, microservices for location data. Each service or configuration can have a different set of input fields. The input fields are displayed after you specify the service and configuration in the Options tab.

The available input fields will change if you change a previously selected service or configuration. If you map input fields and then select a different service or configuration, you must remap the input fields.

For descriptions of the available fields, see the DQM microservices documentation at <https://help.sap.com/viewer/d95546360fea44988eb614718ff7e959/Cloud/en-US>.

8.6.3.4 Output fields for the DQM Microservices transform

The available DQM Microservices transform output fields are determined by the fields available in the service or configuration that has been selected for SAP Data Quality Management, microservices for location data.

Each service or configuration can have a different set of output fields. The output fields are displayed after you specify the service and configuration in the Options tab.

The available output fields will change if you change a previously selected service or configuration. If you map output fields and then select a different service or configuration, you must remap the output fields.

For descriptions of the available fields, see the DQM microservices documentation at <https://help.sap.com/viewer/d95546360fea44988eb614718ff7e959/Cloud/en-US>.

8.6.3.5 Mapping DQM Microservices fields to Data Services fields for the addressCleanse service

The following sections map SAP Data Quality Management, microservices for location data fields to Global Address Cleanse and Geocoder transform fields for use with the addressCleanse service.

If you are using both a Global Address Cleanse or Geocoder transform and a DQM Microservices transform to process your data in the same data flow, add a Query transform downstream from one of the transforms to harmonize the transform output schemas. Then you can add a Merge transform to combine the data sets into a single output data set. For more information, see the *Supplement for SAP*.

8.6.3.5.1 Mapping DQM Microservices input fields to Global Address Cleanse input fields

The following table maps SAP Data Quality Management, microservices for location data input fields to Global Address Cleanse transform input fields for use with the addressCleanse service.

Data Quality Management, microservices for location data input attribute	Global Address Cleanse input field
building	NW_Building
country	Country or NW_Country
floor	NW_Floor_Num
house_num	NW_House_Num1
house_num2	NW_House_Num2
locality	Locality1 or NW_City1
locality2	Locality2 or NW_Home_City
locality3	Locality3 or NW_City2
location	NW_Location
mixed	Multiline1
mixed10	Multiline10

Data Quality Management, microservices for location data input attribute

Global Address Cleanse input field

mixed11	Multiline11
mixed2	Multiline2
mixed3	Multiline3
mixed4	Multiline4
mixed5	Multiline5
mixed6	Multiline6
mixed7	Multiline7
mixed8	Multiline8
mixed9	Multiline9
po_box	NW_PO_Box
po_box_country	NW_PO_Box_Country
po_box_locality	NW_PO_Box_City
po_box_postcode	NW_PO_Box_Postcode
po_box_region	NW_PO_Box_Region
postcode	Postcode or NW_Postcode
region	Region1 or NW_Region
region2	Region2
roomnumber	NW_Room_Num
street	NW_Street
street_suppl	NW_Str_Suppl1
street_suppl2	NW_Str_Suppl2
street_suppl3	NW_Str_Suppl3
suggestionReply	Suggestion_Reply1

8.6.3.5.2 Mapping DQM Microservices output fields to Global Address Cleanse output fields

The following table maps SAP Data Quality Management, microservices for location data output fields to Global Address Cleanse transform output fields for use with the addressCleanse service.

**Data Quality Management,
microservices for location
data generated attribute**

Global Address Cleanse output field

addr_address_rem	Address_Line_Remainder1.Best.Component.Delivery
addr_address_rem2	Address_Line_Remainder2.Best.Component.Delivery
addr_address_rem3	Address_Line_Remainder3.Best.Component.Delivery
addr_address_rem4	Address_Line_Remainder4.Best.Component.Delivery
addr_asmt_info	Assignment_Info.Best.Component.Delivery
addr_asmt_level	Assignment_Level.None.Assignment_Info.None
addr_asmt_type	Assignment_Type.Best.Component.Delivery
addr_change_sig	Change_Significance.None.Assignment_Info.None
addr_extra	Extra1.Best.Component.Delivery
addr_extra2	Extra2.Best.Component.Delivery
addr_extra3	Extra3.Best.Component.Delivery
addr_extra4	Extra4.Best.Component.Delivery
addr_info_code	Info_Code.None.Assignment_Info.None
addr_language	Language.Best.Component.Delivery
addr_po_box_asmt_info	NW_PO_Box_Assignment_Info.Best.Component.Delivery
addr_po_box_asmt_level	NW_PO_Box_Assignment_Level.None.Assignment_Info.None
addr_po_box_asmt_type	NW_PO_Box_Assignment_Type.Best.Component.Delivery
addr_po_box_change_sig	NW_PO_Box_Change_Significance.None.Assignment_Info.None
addr_po_box_info_code	NW_PO_Box_Info_Code.None.Assignment_Info.None
addr_remainder_ex- tra_pmb_full	Remainder_Extra_PMB_Full.Best.Component.Delivery
addr_remainder_full	Remainder_Full.Best.Component.Delivery
addr_sugg_count	Count.None.Suggestion.None
addr_sugg_error	Error.None.Suggestion.None
addr_sugg_list	Suggestion_List.None.Suggestion.None
addr_sugg_status	Status.None.Suggestion.None
sap_formatted_po_box_post- code	NW_PO_Box_NW_Formatted_Postcode.Best.Component.Delivery
sap_formatted_postcode	NW_Formatted_Postcode.Best.Component.Delivery
sap_in_fmt_po_box_postcode	NW_PO_Box_NW_Postcode_In_Supported_Format.Best.Component.Delivery
sap_in_fmt_postcode	NW_Postcode_In_Supported_Format.Best.Component.Delivery
std_addr_additional_info	Additional_Info1.Best.Component.Delivery
std_addr_additional_info2	Additional_Info2.Best.Component.Delivery
std_addr_additional_info3	Additional_Info3.Best.Component.Delivery

**Data Quality Management,
microservices for location**

data generated attribute	Global Address Cleanse output field
std_addr_additional_info4	Additional_Info4.Best.Component.Delivery
std_addr_additional_info5	Additional_Info5.Best.Component.Delivery
std_addr_additional_info6	Additional_Info6.Best.Component.Delivery
std_addr_addr_deldual	Primary_Secondary_Addr_Delivery_Dual.Best.Component.Delivery
std_addr_addr_deldual_bldg	Building_Primary_Secondary_Addr_Delivery_Dual.Best.Component.Delivery
std_addr_address_delivery	Primary_Secondary_Address.Best.Component.Delivery
std_addr_address_dual	Primary_Secondary_Address.Best.Component.Dual
std_addr_area_name	Area_Name1.Best.Component.Delivery
std_addr_block_desc	Block_Description.Best.Component.Delivery
std_addr_block_full	Block_Full.Best.Component.Delivery
std_addr_block_number	Block_Number.Best.Component.Delivery
std_addr_building_name	Building_Name1.Best.Component.Delivery
std_addr_building_name1_2	Building_Name1_2.Best.Component.Delivery
std_addr_building_name2	Building_Name2.Best.Component.Delivery
std_addr_country_2char	ISO_Country_Code_2char.Best.Component.Delivery
std_addr_country_3char	ISO_Country_Code_3char.Best.Component.Delivery
std_addr_country_3digit	ISO_Country_Code_3Digit.Best.Component.Delivery
std_addr_country_name	Country_Name.Best.Component.Delivery
std_addr_delinst_full	Delivery_Installation_Full.Best.Component.Delivery
std_addr_firm	Firm.Best.Component.Delivery
std_addr_floor_desc	Floor_Description.Best.Component.Delivery
std_addr_floor_full	Floor_Full.Best.Component.Delivery
std_addr_floor_number	Floor_Number.Best.Component.Delivery
std_addr_floor_qual	Floor_Qualifier.Best.Component.Delivery
std_addr_full_address	Full_Address.Best.Component.Delivery
std_addr_lastline	Lastline.Best.Component.Delivery
std_addr_locality	Locality1_Name.Best.Component.Delivery
std_addr_locality_desc	Locality1_Description.Best.Component.Delivery
std_addr_locality_full	Locality1_Full.Best.Component.Delivery
std_addr_locality1_2	Locality1_2_Name.Best.Component.Delivery
std_addr_locality1_2_full	Locality1_2_Full.Best.Component.Delivery
std_addr_locality1_4	Locality1_4_Name.Best.Component.Delivery
std_addr_locality1_4_full	Locality1_4_Full.Best.Component.Delivery

**Data Quality Management,
microservices for location**

data generated attribute

Global Address Cleanse output field

std_addr_locality2	Locality2_Name.Best.Component.Delivery
std_addr_locality2_4	Locality2_4_Name.Best.Component.Delivery
std_addr_locality2_4_full	Locality2_4_Full.Best.Component.Delivery
std_addr_locality2_desc	Locality2_Description.Best.Component.Delivery
std_addr_locality2_full	Locality2_Full.Best.Component.Delivery
std_addr_locality3	Locality3_Name.Best.Component.Delivery
std_addr_locality3_4	Locality3_4_Name.Best.Component.Delivery
std_addr_locality3_4_full	Locality3_4_Full.Best.Component.Delivery
std_addr_locality3_desc	Locality3_Description.Best.Component.Delivery
std_addr_locality3_full	Locality3_Full.Best.Component.Delivery
std_addr_locality4	Locality4_Name.Best.Component.Delivery
std_addr_locality4_desc	Locality4_Description.Best.Component.Delivery
std_addr_locality4_full	Locality4_Full.Best.Component.Delivery
std_addr_pmb_full	PMB_Full.Best.Component.Delivery
std_addr_pname_secaddr	PName_Secondary_Addr.Best.Component.Delivery
std_addr_po_box_country_2char	NW_PO_Box_ISO_Country_Code_2char.Best.Component.Delivery
std_addr_po_box_delinst_full	NW_PO_Box_Delivery_Installation_Full.Best.Component.Delivery
std_addr_po_box_full	NW_PO_Box_Primary_Address.Best.Component.Delivery
std_addr_po_box_locality_full	NW_PO_Box_Locality1_Full.Best.Component.Delivery
std_addr_po_box_number	NW_PO_Box_Primary_Number.Best.Component.Delivery
std_addr_po_box_postcode_full	NW_PO_Box_Postcode_Full.Best.Component.Delivery
std_addr_po_box_region	NW_PO_Box_Region1.Best.Component.Delivery
std_addr_po_box_region_code	NW_PO_Box_Region1_Code.Best.Component.Delivery
std_addr_po_box_region_full	NW_PO_Box_Region1_Full.Best.Component.Delivery
std_addr_po_box_region2	NW_PO_Box_Region2.Best.Component.Delivery
std_addr_po_box_region2_code	NW_PO_Box_Region2_Code.Best.Component.Delivery
std_addr_po_box_region2_full	NW_PO_Box_Region2_Full.Best.Component.Delivery
std_addr_point_of_ref	Point_Of_Reference1.Best.Component.Delivery
std_addr_point_of_ref1_2	Point_Of_Reference1_2.Best.Component.Delivery
std_addr_point_of_ref2	Point_Of_Reference2.Best.Component.Delivery
std_addr_postcode_full	Postcode_Full.Best.Component.Delivery

**Data Quality Management,
microservices for location**

data generated attribute

Global Address Cleanse output field

std_addr_postcode1	Postcode1.Best.Component.Delivery
std_addr_postcode2	Postcode2.Best.Component.Delivery
std_addr_prim_address	Primary_Address.Best.Component.Delivery
std_addr_prim_name	Primary_Name1.Best.Component.Delivery
std_addr_prim_name_full	Primary_Name_Full1.Best.Component.Delivery
std_addr_prim_name1_2	Primary_Name_Full1_2.Best.Component.Delivery
std_addr_prim_name1_4	Primary_Name_Full1_4.Best.Component.Delivery
std_addr_prim_name2	Primary_Name2.Best.Component.Delivery
std_addr_prim_name2_full	Primary_Name_Full2.Best.Component.Delivery
std_addr_prim_name3	Primary_Name3.Best.Component.Delivery
std_addr_prim_name3_4	Primary_Name_Full3_4.Best.Component.Delivery
std_addr_prim_name3_full	Primary_Name_Full3.Best.Component.Delivery
std_addr_prim_name4	Primary_Name4.Best.Component.Delivery
std_addr_prim_name4_full	Primary_Name_Full4.Best.Component.Delivery
std_addr_prim_number	Primary_Number.Best.Component.Delivery
std_addr_prim_number_desc	Primary_Number_Description.Best.Component.Delivery
std_addr_prim_number_extra	Primary_Number_Extra.Best.Component.Delivery
std_addr_prim_number_full	Primary_Number_Full.Best.Component.Delivery
std_addr_prim_postfix	Primary_Postfix1.Best.Component.Delivery
std_addr_prim_prefix	Primary_Prefix1.Best.Component.Delivery
std_addr_prim_type	Primary_Type1.Best.Component.Delivery
std_addr_prim_type2	Primary_Type2.Best.Component.Delivery
std_addr_prim_type3	Primary_Type3.Best.Component.Delivery
std_addr_prim_type4	Primary_Type4.Best.Component.Delivery
std_addr_primaddr_deldual	Primary_Address_Delivery_Dual.Best.Component.Delivery
std_addr_primaddr_deldual_bldg	Building_Primary_Addr_Delivery_Dual.Best.Component.Delivery
std_addr_region	Region1.Best.Component.Delivery
std_addr_region_code	Region1_Code.Best.Component.Delivery
std_addr_region_full	Region1_Full.Best.Component.Delivery
std_addr_region1_2	Region1_2_Name.Best.Component.Delivery
std_addr_region1_2_full	Region1_2_Full.Best.Component.Delivery
std_addr_region2	Region2.Best.Component.Delivery

**Data Quality Management,
microservices for location**

data generated attribute	Global Address Cleanse output field
std_addr_region2_code	Region2_Code.Best.Component.Delivery
std_addr_room_full	Room_Full.Best.Component.Delivery
std_addr_room_number	Room_Number.Best.Component.Delivery
std_addr_sec_address	Secondary_Address.Best.Component.Delivery
std_addr_secaddr_no_floor	Secondary_Address_No_Floor.Best.Component.Delivery
std_addr_se- caddr_no_floor_room	Secondary_Address_No_Floor_No_Room.Best.Component.Delivery
std_addr_secaddr_no_room	Secondary_Address_No_Room.Best.Component.Delivery
std_addr_single_address	Single_Address.Best.Component.Delivery
std_addr_stairwell_desc	Stairwell_Description.Best.Component.Delivery
std_addr_stairwell_full	Stairwell_Full.Best.Component.Delivery
std_addr_stairwell_name	Stairwell_Name.Best.Component.Delivery
std_addr_unit_desc	Unit_Description.Best.Component.Delivery
std_addr_unit_full	Unit_Full.Best.Component.Delivery
std_addr_unit_number	Unit_Number.Best.Component.Delivery
std_addr_unit_qual	Unit_Qualifier.Best.Component.Delivery
std_addr_wing_desc	Wing_Description.Best.Component.Delivery
std_addr_wing_full	Wing_Full.Best.Component.Delivery
std_addr_wing_name	Wing_Name.Best.Component.Delivery
sugg_addr_address_delivery (in addr_sugg_list)	Sugg_Full_Addressline.None.Suggestion.None
sugg_addr_lastline (in addr_sugg_list)	Sugg_Full_Lastline.None.Suggestion.None
sugg_addr_prim_number_high (in addr_sugg_list)	Primary_Number_High.None.Suggestion.None
sugg_addr_prim_number_low (in addr_sugg_list)	Primary_Number_Low.None.Suggestion.None
sugg_addr_prim_side_indica- tor (in addr_sugg_list)	Primary_Side_Indicator.None.Suggestion.None
sugg_addr_sec_side_indicator (in addr_sugg_list)	Secondary_Side_Indicator.None.Suggestion.None
sugg_addr_selection (in addr_sugg_list)	Selection.None.Suggestion.None
sugg_addr_single_address (in addr_sugg_list)	Sugg_Single_Address.None.Suggestion.None

**Data Quality Management,
microservices for location**

data generated attribute **Global Address Cleanse output field**

sugg_addr_unit_number_high (in addr_sugg_list)	Unit_Number_High.None.Suggestion.None
---	---------------------------------------

sugg_addr_unit_number_low (in addr_sugg_list)	Unit_Number_Low.None.Suggestion.None
--	--------------------------------------

8.6.3.5.3 Mapping DQM Microservices output fields to Geocoder output fields

The following table maps SAP Data Quality Management, microservices for location data output fields to Geocoder transform output fields for use with the addressCleanse service.

Data Quality Management, microservices for location

data generated attribute **Geocoder output field**

addr_latitude	Latitude
addr_longitude	Longitude
geo_info_code	Info_Code
geo_asmt_level	Assignment_Level

8.6.3.6 Mapping DQM Microservices fields to Data Services fields for the reverseGeo service

The following sections map SAP Data Quality Management, microservices for location data fields to Geocoder transform fields for use with the reverseGeo service.

8.6.3.6.1 Mapping DQM Microservices input fields to Geocoder input fields

The following table maps SAP Data Quality Management, microservices for location data input fields to Geocoder transform input fields for use with the reverseGeo service.

Data Quality Management, microservices for location**data input attribute****Geocoder input field**

latitude	Latitude
longitude	Longitude

8.6.3.6.2 Mapping DQM Microservices output fields to Geocoder output fields

The following table maps SAP Data Quality Management, microservices for location data output fields to Geocoder transform output fields for use with the reverseGeo service.

Data Quality Management, microservices for location**data generated attribute****Geocoder output field**

geo_info_code	Info_Code
geo_search_results	Result_List
geo_search_results_count	Result_List_Count
search_geo_addr_latitude (in geo_search_results)	Latitude
search_geo_addr_longitude (in geo_search_results)	Longitude
search_geo_address_delivery (in geo_search_results)	Address_Line
search_geo_asmt_level (in geo_search_results)	Assignment_Level
search_geo_country_2char (in geo_search_results)	Country_Code
search_geo_distance (in geo_search_results)	Distance
search_geo_group_id (in geo_search_results)	Group_Number
search_geo_group_master (in geo_search_results)	Group_Rand
search_geo_locality (in geo_search_results)	Locality1
search_geo_locality2 (in geo_search_results)	Locality2
search_geo_poi_name (in geo_search_results)	POI_Name
search_geo_poi_type (in geo_search_results)	POI_Type
search_geo_population_class (in geo_search_results)	Population_Class_Locality1
search_geo_postcode_full (in geo_search_results)	Postcode
search_geo_region (in geo_search_results)	Region1
search_geo_side_of_street (in geo_search_results)	Side_Of_Primary_Address

8.6.4 Map_Operation



Modifies data based on mapping expressions and current operation codes. The operation codes can be converted between data manipulation operations.

i Note

If you configure both mapping expressions and operation codes, Data Services runs mapping expressions before any row type conversions. In addition, mapping expressions are optional, so if you don't have any configured Data Services performs operation codes only.

Writing map expressions per column and per row type (INSERT/UPDATE/DELETE) allows you to:

- change the value of data for a column.
- execute different expressions on a column, based on its input row type.
- use the `before_image` function to access the before image value of an UPDATE row.

i Note

You can use mapping expressions on only top-level schemas. They do not work on nested schemas.

This transform can also change operation codes on data sets to produce the desired output. For example, if a row in the input data set has been updated in some previous operation in the data flow, you can use this transform to map the UPDATE operation to an INSERT. The result of converting UPDATE rows into INSERT rows is the preservation the rows in the target.

Data Services can push Map_Operation transforms to the source database.

Parent topic: [Platform transforms \[page 833\]](#)

Related Information

[Case \[page 833\]](#)

[Data Mask \[page 836\]](#)

[DQM Microservices \[page 925\]](#)

[Merge \[page 942\]](#)

[Query \[page 943\]](#)

[Row_Generation \[page 970\]](#)

[SQL \[page 972\]](#)

[User-Defined \[page 975\]](#)

[Validation \[page 983\]](#)

[XML_Map \[page 994\]](#)

[Effective_Date \[page 375\]](#)

[real \[page 313\]](#)

[before_image \[page 1052\]](#)

8.6.4.1 Data inputs

A data set with rows flagged with any operation codes.

The input data set can contain hierarchical data.

Use caution when using columns of data type `real` in this transform. Comparison results are unpredictable for this data type.

8.6.4.2 Options

Tab	Option	Description
Mapping	Update mapping	Allows you to enter and edit optional mapping expressions for output columns and row types.
	Insert/Normal mapping	
	Delete mapping	
Map Operations	Output row type	Indicate the new operations desired for the input data set. Choose from the following operation codes: INSERT, UPDATE, DELETE, NORMAL, or DISCARD.

8.6.4.3 Data outputs

A data set with rows flagged as specified by the mapping operations.

Rows in the input data set can contain any of the following operation codes:

- NORMAL
- INSERT
- DELETE
- UPDATE

Any of these operation codes can be mapped to:

- NORMAL
- INSERT
- DELETE
- UPDATE

In addition, the DISCARD option can be assigned. Discarded rows are not passed through to the output of the transform.

By default, every input operation type maps to itself. For each specified mapping, every row in the input data set that matches the input mapping operation is converted to the specified output operation.

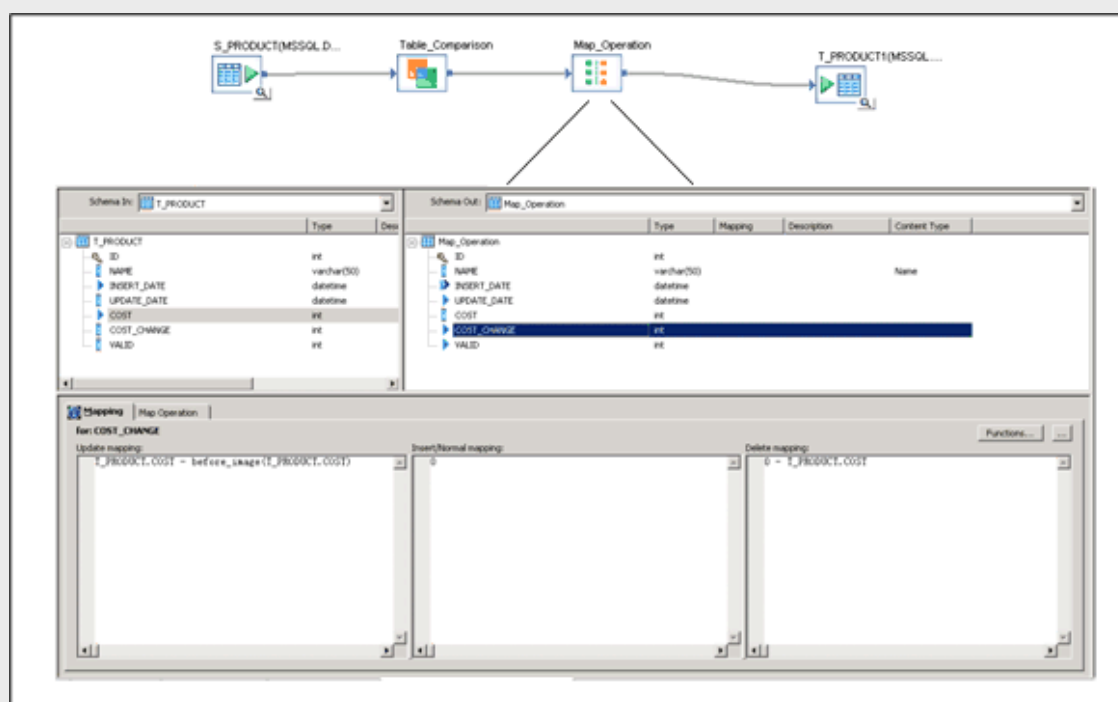
❖ Example

Showing a change in cost

In this example, the original target table shows the cost field value as 9,000 and the cost_change field value as 0. Later on, the input data changes so that the cost field value is 1,000.

To see what the change in cost is after the input changes, you would do the following:

1. Set up the Table_Comparison transform to determine the row type (for example, INSERT or UPDATE).
2. In the Map_Operation transform, the Mapping tab would contain an expression for each column and each row type.
 - For the INSERT row type, fill the COST_CHANGE field with 0.
 - For the UPDATE row type, the before image value of the COST field (before_image(T_PRODUCT1.COST)) would be subtracted from the updated value of the COST field (T_PRODCT1.COST). This gives you a COST_CHANGE field value of negative 8000 (1,000 - 9,000 = -8,000).
 - For the DELETE row type, the COST_CHANGE field is set as 0 - T_PRODUCT1.COST.



3. The Map Operation tab would have the following settings:

Input row type	Output row type
normal	normal
update	update

Input row type	Output row type
insert	insert
delete	update

❖ Example

Changing a field value and a row type

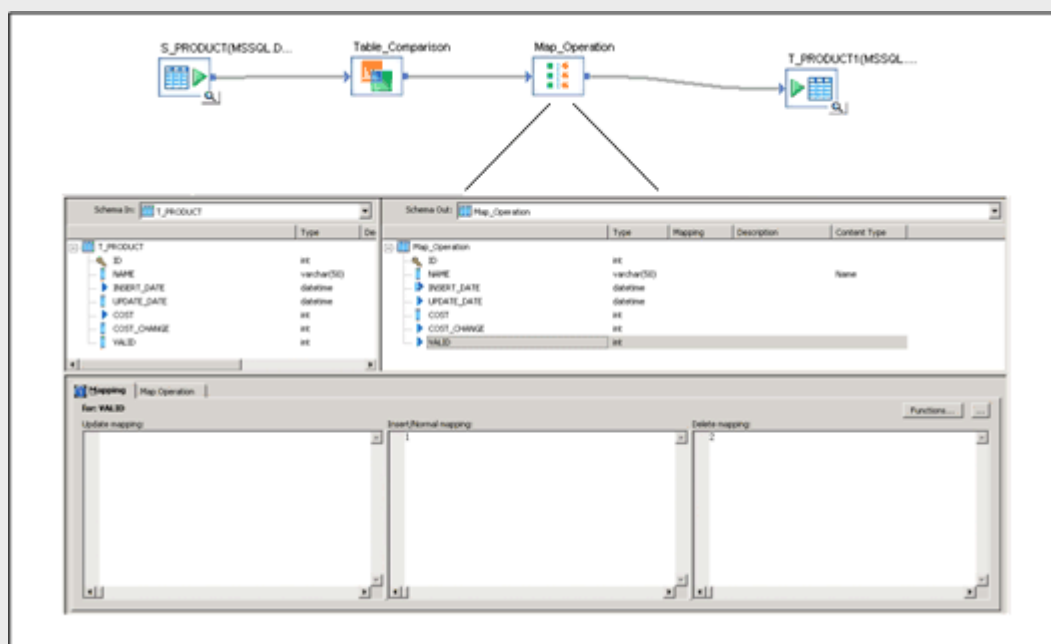
In this example, an expression is created to set the VALID field to 0 and to convert the DELETE row type to an UPDATE row type.

The VALID field (1=valid, 0=invalid) indicates if a product is in production or not. You do not want to delete the product from the target even though it has been deleted from the source. Instead, you want to update the valid field in the target to 0.

i Note

When configuring both mapping expressions and operation codes, Data Services runs mapping expressions before any row type conversions.

The Mapping tab would appear as follows:



The Map Operation tab would have the following settings:

Input row type	Output row type
normal	normal
update	update

Input row type	Output row type
insert	insert
delete	update

Related Information

[before_image](#) [page 1052]

[Table_Comparison](#) [page 422]

8.6.5 Merge



Combines incoming data sets, producing a single output data set with the same schema as the input data sets.

Parent topic: [Platform transforms](#) [page 833]

Related Information

[Case](#) [page 833]

[Data Mask](#) [page 836]

[DQM Microservices](#) [page 925]

[Map_Operation](#) [page 938]

[Query](#) [page 943]

[Row_Generation](#) [page 970]

[SQL](#) [page 972]

[User-Defined](#) [page 975]

[Validation](#) [page 983]

[XML_Map](#) [page 994]

8.6.5.1 Data inputs

A data set from two or more sources with rows flagged as any operation code.

All sources must have the same schema, including:

- The same number of columns
- The same column names
- The same data types of columns

If the input data set contains hierarchical data, the names and data types must match at every level of the hierarchy.

8.6.5.2 Options

None.

8.6.5.3 Data outputs

A data set consisting of rows from all sources, with any operation codes. The output data has the same schema as the source data, including nested schemas.

The output data set contains a row for every row in the source data sets. The transform does not strip out duplicate rows. If columns in the input set contain nested schemas, the nested data is passed through without change.

If the data types of columns in the sources do not match the target, add a query in the data flow before the Merge transform. In the query, apply a data type conversion to the columns with data types that do not match the target column data types.

You must apply other operations such as DISTINCT in a query following the Merge transform.

8.6.6 Query

The Query transform retrieves a data set that satisfies conditions that you specify. This transform is similar to a SQL `SELECT` statement.



The Query transform can perform the following operations:

- Choose (filter) the data to extract from sources
- Join data from multiple sources
- Map columns from input to output schemas
- Perform transformations and functions on the data
- Perform data nesting and unnesting

- Add new columns, nested schemas, and function results to the output schema
- Assign primary keys to output columns

Parent topic: [Platform transforms \[page 833\]](#)

Related Information

[Case \[page 833\]](#)

[Data Mask \[page 836\]](#)

[DQM Microservices \[page 925\]](#)

[Map_Operation \[page 938\]](#)

[Merge \[page 942\]](#)

[Row_Generation \[page 970\]](#)

[SQL \[page 972\]](#)

[User-Defined \[page 975\]](#)

[Validation \[page 983\]](#)

[XML_Map \[page 994\]](#)

[Query transform output schema \[page 947\]](#)

[Query transform input schema \[page 946\]](#)

[Smart editor \[page 299\]](#)

8.6.6.1 Data inputs and outputs

A Query transform takes data from upstream objects in the data flow and sends data to downstream objects in the data flow.

Data that is input to a Query transform can be from one source, or many sources. You control the data that is output by using the various features of the Query transform. You view and manipulate the data using the *Schema In* and *Schema Out* panes in the Query transform.

i Note

If you aren't familiar with the many things you can do with your data using the Query transform, see the Data services *Tutorial* for more information.

i Note

Use caution when using columns of data type `real` in this transform. Comparison results are unpredictable for the `real` data type.

8.6.6.2 Editor

Use the Query editor to specify input and output schemas, and to control the data using SQL expressions.

The Query Editor contains the following areas:

- *Schema In* pane in the upper left
- *Schema Out* pane in the upper right
- *Parameters* pane in the lower area

The *Schema In* and *Schema Out* panes can contain columns, nested schemas, and functions (output only).

The areas can be resized in order to expand the area in which you are working. You can also expand and contract the columns to change the width of properties displayed in the input and output schema areas.

8.6.6.3 Searching in an input or output schema

1. In the Query editor Find tab, enter the search term in the *Find what* box or select from previous search terms in the drop-down list.
2. In the *Schemas* list, choose the schemas in which to search.
3. In the *Elements* list, choose the types of mappings in which to search.
4. In the *Where* list, choose the properties to search within.

i Note

You can search within one or all properties, but not within two or three specific properties at a time.

5. Select the *Match case* check box to constrain your search to the capitalization entered.
6. Click *Find*.
The Designer searches the query for the words you specified within the constraints you defined.

i Note

The Designer searches for columns loaded into memory. If columns are not loaded into memory, you must expand the schema to load the columns into memory before clicking *Find* and searching for the columns.

All matches are shown in the box below the find constraints. When you click to select a table or column name, the table or column is automatically highlighted and shown in the corresponding input or output schema area.

Initially, the Designer lists the matching columns in the order that they appear within the schemas. If you are searching both schemas, the Designer lists the first match found in the input schema first and the last match found in the output schema last. You can sort the list of matches by property. Each time you click a property heading, the Designer resorts the matches, cycling through original order, ascending order, and descending order.




Arrow icons confirm column and sort type. For example, if you sort the data by the *Description* property and in ascending order, an "up" arrow appears next to the *Description* heading. Click the heading again and a "down" arrow appears to indicate that the data is now sorted in descending alpha-numeric order. Click again and the match list returns to its original sort order.

8.6.6.4 Query transform input schema

The input schema area displays all schemas input to the Query transform as a hierarchical tree. Each input schema can contain zero or more of the following elements:

- Columns
- Nested schemas

Icons preceding columns are combinations of the following graphics:

Icon	Description
	Primary key.
	Column that is not used in output mapping.
	Column that is used in output mappings.

The *Input* list at the top left of the query editor indicates the schema that is currently selected. As you select schemas or columns in the input schema area, the *Input* list displays the corresponding schema. Conversely, you can select a schema in the *Input* list to move easily to a required schema.

You can right-click elements in the input schema area and select the following menu commands:

Command	Applicable elements	Effect
Copy	Columns, schemas	Stores a copy of the selected elements in the clipboard, leaving the elements in the input schema area.
Find	Anywhere in input schema area	Locates an output element with the name or description you enter.
Refresh	Anywhere in input schema area	Refreshes the display of the input schema area.
Parent	Columns	Selects the parent schema of the selected column.
Collapse	Columns, schemas	Collapses a selected schema or a selected column's parent schema (to facilitate viewing/navigation).
Generate DTD...		Generates a DTD format that corresponds to the structure of the selected schema (either NRDM or relational). Generates all data types as varchar.
Generate XML Schema...		Generates an XML Schema that corresponds to the structure of the selected schema (either NRDM or relational). All data types match those of the selected schema.





Command	Applicable elements	Effect
Propagate Column From...	Columns	<p>Carries a selected column schema from an upstream source or transform through intermediate objects to the input schema.</p> <p>Simple mappings are created in each object with no change to the data type or data itself.</p>
Map to Output	Columns, schemas	Creates a simple mapping from the input schema area to the output schema area.
Create File Format...		Creates a file format from a relational table schema. All data types match those of the original table schema.
Properties	Columns, schemas	<p>Displays the properties of the selected element.</p> <p>You cannot modify the properties.</p>

8.6.6.5 Query transform output schema

The output schema area displays the schema output from the Query transform as a hierarchical tree. The output schema can contain one or more of the following elements:

- Columns
- Nested schemas
- Functions

Icons preceding columns are combinations of the following graphics:

Icon	Description
	Primary key.
	Column that has a simple mapping. A simple mapping is either a single column or an expression with no input column (that is, an expression that does not vary with input).
	Column that has a complex mapping. A complex mapping is any mapping that is not simple.
	(Red cross superimposed on any icon) Incorrect mapping. Data Services does not perform a complete validation during design, so the editor may not flag an incorrect mapping. For a complete validation, select Validation > Validate .

The Schema Out pane shows the following:

- The current schema in the Schema Out list at the top and in the output schema area. The current schema determines:
 - The output elements that you can modify (add, map, or delete).
 - The scope of the SELECT through ORDER BY tabs in the options area.
- Non-current schemas appear dim in the output schema area.

Related Information

[Schema In and Schema Out \[page 948\]](#)

8.6.6.6 Schema In and Schema Out

The *Schema In* and *Schema Out* panes determine the fields that continue on in your data flow as well as field mapping.

The software helps you populate the schema panes in the Query transform based on objects and connections in your data flow. You can switch the focus in the *Query Editor* to change the current schema. You can also automatically complete the *Schema Out* pane with a target table schema.

There are several ways to change the current schema in a Query transform:

- Select a schema in the *Schema Out* pane.
- Right-click a schema, column, or function in the *Schema Out* pane and select *Make Current*.
- Double-click a schema in the *Schema Out* pane.

To automatically complete an empty output schema with a target table schema, connect the Query transform to a target table in your data flow. If the *Schema Out* pane is already populated, the software does not overwrite those mappings.

There are several techniques to map an input schema to an output schema:

- Drag and drop columns or nested schemas from the input schema area to the output schema area for simple column mappings. When you drop a single column on top of an existing column, you can remap that column.
 - Select *Remap Column* to update only the column mapping.
 - Select *Remap with Data Type* to update the column mapping a data type.
 - Select *Insert Above* or *Insert Below* to add a column as a new mapping.
- Right-click the current schema in the *Schema Out* pane and select *New Output Column* or *New Output Schema*. You can provide simple column mappings by dragging input columns over the new output columns. For complex mappings, use the options area.
- Right-click columns in the current schema in the *Schema Out* pane to assign and reverse primary key settings. A key icon indicates primary keys.
- Right-click the current schema in the *Schema Out* pane and select *New Function*.
 - Only functions that are in the repository are available.
 - Add adapter functions and SAP RFC or BAPI functions. These types of functions return multiple columns (in contrast to the functions used in mappings and Where clauses, which return single values).

- Map all first-level input parameters for the function to the input parameters of the query.
- Right-click the current schema in the *Schema Out* pane and select *Unnest* to flatten nested output schemas. Use the *Unnest* option when the source has a nested schema, such as an XML file. You map columns from this source to a flat target table schema.
- Right-click elements in the *Schema Out* pane and select a command. The elements must be within the current schema.

Related Information

[Query transform output schema \[page 947\]](#)

[Output schema commands \[page 949\]](#)

8.6.6.6.1 Output schema commands

When you right-click an element such as a column in the *Schema Out* pane of the Query transform, you can choose from several commands to control output content.


The following table contains descriptions for the commands available to select.

Command	Applicable elements	Effect
<i>Cut</i>	All	Removes the selected elements from the <i>Schema Out</i> pane and stores a copy of the elements in the clipboard.
<i>Copy</i>	All	Stores a copy of the selected elements in the clipboard, leaving the elements in the <i>Schema Out</i> pane.
<i>Paste</i>	All	<p>Inserts the elements stored in the clipboard at the current cursor location (this must be within the current schema). Only visible when the clipboard contains something.</p> <p>If the cursor overlaps an existing column, you are prompted to insert above, insert below, remap column, or cancel.</p>
<div> <div>i Note</div> <p>Copying an input element and pasting it in the <i>Schema Out</i> pane provides a simple mapping from the input element to the output element. You can also do this by dragging the input element to the <i>Schema Out</i> pane.</p> </div>		
<i>Delete</i>	All	Removes the selected elements from the <i>Schema Out</i> pane, without making a copy.
<i>Find...</i>	All	Locates an output element with the name or description you enter.

Command	Applicable elements	Effect
<i>Make Current</i>	All except for the current schema	Makes the selected schema, or the schema of the selected element, the current schema.
<i>New Output Column...</i>	Schemas	Adds an output column to the current schema with the name and properties you enter.
<i>New Output Schema...</i>	Schemas	Adds a nested schema to the current schema with the name you enter.
<i>New Function Call</i>	Schemas	<p>Adds a function or stored procedure call to the current schema. The function or procedure must already be imported into the repository.</p> <p>You can add adapter functions and SAP RFC or BAPI functions. These functions return multiple columns (in contrast to the functions used in mappings and Where clauses that return single values).</p> <p>In the <i>Define Input Parameter(s)</i> window, map all first-level input parameters for the function to the input parameters of the query.</p>
<i>Modify Function Call</i>	Functions	<p>Allows you to modify the selected function.</p> <p>In the <i>Define Input Parameter(s)</i> window, map all first-level input parameters for the function to the input parameters of the query.</p>
<i>Propagate Column From</i>	Columns	<p>Carries a selected column schema from an upstream source or transform through intermediate objects to the output schema.</p> <p>Simple mappings are created in each object with no change to the data type or data itself.</p>
<i>Unnest</i>	Nested schemas	Toggles to flatten or re-nest the selected schema into the parent schema in the query output. An unnested schema will not appear in the succeeding transform or target; only its columns appear. Unnested schemas appear in the Query transform <i>Schema Out</i> pane as table icons with a black arrow that points to the left.
<i>Nest with sub-schemas</i>	Nested schemas	Re-nests the selected schema and all sub-schemas into the parent schema in the query output.
<i>Unnest with sub-schemas</i>	Nested schemas	Flattens the selected schema and all sub-schemas into the parent schema in the query output. Unnested schemas and sub-schemas do not appear in the succeeding transform or target; only the columns appear. Unnested schemas and sub-schemas appear in the Query transform <i>Schema Out</i> pane as table icons with a black arrow that points to the left.
<i>Primary Key</i>	Columns	Toggles the primary key attribute of the column on (check mark appears next to the command) or off (no check mark appears next to the command). A key icon indicates that a column is a primary key.

Command	Applicable elements	Effect
<i>Optional</i>	Schemas	Toggles to make a schema optional.
<i>Generate DTD</i>	All	Generates a DTD format that corresponds to the structure of the selected schema (either NRDM or relational). Generates all data types as varchar.
<i>Generate XML Schema</i>	Schema	Generates an XML Schema that corresponds to the structure of the selected schema (either NRDM or relational). All data types match those of the selected schema.
<i>Create File Format</i>	Schema	Creates a file format from a relational table schema. All data types match those of the original table schema.
<i>Properties</i>	All	Displays the properties of the selected element.

8.6.6.7 Options

The options area of the Query Editor contains several tabs where you enter information to specify the data you want retrieved. Specifying information on these tabs is similar to specifying a SQL SELECT statement. Tabs containing entries are flagged by a special  icon.

When you drag and drop (or copy and paste) input columns to the output schema, Data Services inserts values in the Mapping and FROM tabs. For simple mappings, this is often sufficient.

For more complex mappings, complete the appropriate tabs.

Query Editor tab descriptions

Tab	Description
Mapping	Specifies how the selected output column will be derived (or mapped).
SELECT	Specifies distinct rows to output (discarding any identical duplicate rows).
FROM	Lists all input schemas. Allows you to specify join pairs and join conditions as well as enter join rank and cache for each input schema. The resulting SQL FROM clause is displayed.
WHERE	<p>Specifies conditions that determine which rows are output.</p> <p>Enter the conditions in SQL syntax, like a WHERE clause in a SQL SELECT statement. For example:</p> <pre>TABLE1.EMPNO = TABLE2.EMPNO AND TABLE1.EMPNO > 1000 OR TABLE2.EMPNO < 9000</pre> <p>Use the Functions, Domains, and smart editor buttons for help building expressions.</p>
GROUP BY	Specifies how the output rows are combined (if required).

Tab	Description
ORDER BY	Specifies how the output rows are sorted (if required).
Advanced	Creates separate sub data flows to process resource-intensive query clauses.
Find	Enables you to search for a specific word or term in the input and/or output schemas.

i Note

Use the SELECT through ORDER BY tabs to specify additional constraints for the current schema, similar to SQL SELECT statement clauses.

Related Information

[Mapping tab \[page 952\]](#)

[SELECT tab \[page 954\]](#)

[FROM tab \[page 954\]](#)

[WHERE tab \[page 955\]](#)

[GROUP BY tab \[page 956\]](#)

[ORDER BY tab \[page 956\]](#)

[Advanced tab \[page 957\]](#)

[Find tab \[page 957\]](#)

8.6.6.8 Query Editor tabs

8.6.6.8.1 Mapping tab

Use the Mapping tab to specify how the selected output column is derived (or mapped). You can specify any valid expression.

Most commonly, mapping expressions contain table columns and functions.

- Enter input column names or drag columns from the input schema and drop them in the box on the Mapping tab.
- Insert functions by entering them directly, using the smart editor, or by clicking the Functions button to open the function wizard.

i Note

You cannot add comments to a mapping clause in a Query transform. For example, the following syntax is not supported in the Mapping tab:

```
table.column # comment
```

If you add comments, the job will not run and you cannot successfully export it. Use the object description or workspace annotation feature instead.

After you map your source to the Query transform, you might determine that you need to use another transform before you send the data to the Query transform. For example, you might add a validation transform to ensure that only data with a certain format is passed or you might add a Case transform to send only a subset of the data.

In general, when you change an input schema to the Query transform, the Designer checks the existing top-level mappings to determine if any remapping is required.

- If the mapping contains a column with a table name that is not a current input schema name and the column is in the new input schema, the Designer automatically replaces the table name with the new input schema name. Specifically, the Designer automatically updates the input schema name for each matching column in the following option tabs of the Query editor:
 - Mapping
 - FROM
 - WHERE
 - ORDER BY
 - GROUP BY
- If the mapping contains a column that was in the obsolete table, but the column does not exist in the new input schema, you must either remove the column or remap it from the original source.

The Designer does not automatically remap the input schema for the following situations:

- When you connect a new source to the Query transform before you disconnect the old source. You must click the [Schema Remapping](#) button on the Mapping tab to update the input schema name for columns in the Mapping, WHERE, GROUP BY, and ORDER BY tabs.
- When the source is a nested schema and you either change the source to a similar nested schema, or you add or delete a transform before the Query transform. Click the [Schema Remapping](#) button to update the Mapping input schema name.

8.6.6.8.1.1 Remapping when automatic remapping was not done in the Query transform

1. In the Mapping tab, click the [Schema Remapping](#) button. The [Replace Obsolete Schema window](#) opens.
2. In the [Specify obsolete schema](#) drop-down list, choose the source schema that you disconnected from the Query transform.
This list displays only the top-level input schema. For an obsolete nested schema, enter the name of the top-level schema.
3. In the [Choose correct schema](#) list, choose the output schema of the transform that you added between the source and Query transform.
4. Click [Remap](#). A message displays the number of columns that were remapped; for example:

```
Schema "ODS_SALESORDER" was replaced by schema "Validation_Pass" in 11 column names.
```

8.6.6.8.2 SELECT tab

Use the SELECT tab to output only distinct rows (discarding any identical duplicate rows).

To discard duplicate rows, select the *Distinct rows* check box. This is similar to specifying a SELECT DISTINCT SQL statement.


8.6.6.8.3 FROM tab

Use the FROM tab to specify input schemas as well as join information, such as join pairs, join type, and the join condition used in the current output schema. The specified information is similar to the FROM clause in a SQL SELECT statement.

The *FROM* tab is divided into three areas:

- The uppermost area contains information about the source tables connected to the Query transform in the data flow. The Input schemas column is populated with the names of the source tables.
- The middle area, *Join pairs*, allows you to specify the tables to be joined, the type of join, and the join condition.
- The lower area, *FROM clause*, displays the SQL FROM clause and is automatically updated as you add join conditions. This area is read-only, but can be copied to the clipboard.

The following table describes the columns displayed in the *Join pairs* area.

Column name	Description
Left	The left source of a join. For the first join pair, select the input schema from the drop-down list of available schemas. For subsequent join pairs, the result of the previous join pair is taken as the left source and the schema cannot be specified.
Join Type	The type of join. Valid values are Inner join and Left outer join. If a table is not explicitly joined to another table, then it is cross-joined (Cartesian Product) to the result of the final join defined by the join pairs.
Right	The right source of a join. Any input schema not used in a previous join pair.
smart editor	Optional: click the icon to open the smart editor. Within the smart editor <i>Data</i> tab, you can drag and drop columns to specify the join condition.
...	
Propose Join	Optional: Click <i>Propose Join</i> to have Data Services generate a join expression.
	The SQL clause is automatically updated after you change the left source, right source, or join type.

Column name	Description
Join condition	<p>A join condition is required for each join pair.</p> <p>Where possible, Designer automatically suggests a join condition based on the input schemas of the join pair. To edit the join condition, you can enter the join condition field or use the smart editor.</p>

Note

If your expression contains varchar comparisons, Data Services ignores trailing blanks in the data. For Oracle data, use the rtrim or rpad functions if the number of trailing blanks might differ on either side of the comparison.

8.6.6.8.4 WHERE tab

Use the WHERE tab to set conditions that determine which rows are output. Enter the conditions in SQL syntax, as you would a WHERE clause in a SQL SELECT statement. The [Propose](#) button generates possible join conditions. You can specify the following conditions:

Condition	Example
Data set filters	<code>TABLE1.EMPNO > 1000</code>
Multiple conditions using logical operators	<code>TABLE1.EMPNO > 1000 OR TABLE2.EMPNO < 9000</code>
Join conditions for inner joins only	<code>TABLE1.EMPNO = TABLE2.EMPNO</code>

Note

Although it is technically possible to specify inner join conditions in the WHERE tab, the best practice is to specify inner join conditions in the FROM tab.

You can specify any valid expression. To enter conditions, do one of the following:

- Enter expressions in the editor.
- Drag columns from the input schema area to the editor.
- Use the Functions button. Use the pushdown_sql function to have Data Services create WHERE clauses dynamically based on data rather than pre-specifying the clause.

i Note

If your expression contains varchar comparisons, Data Services ignores trailing blanks in the data. For Oracle data, use the `rtrim` or `rpadd` functions if the number of trailing blanks might differ on either side of the comparison.

8.6.6.8.5 GROUP BY tab

Use the GROUP BY tab to specify a list of columns for which you want to combine output. For each unique set of values in the group by list, Data Services combines or aggregates the values in the remaining columns. For example, you might want to group sales order records by order date to find the total sales ordered on a particular date.

To add a column to the Group By list, select the column in the input schema area and drag it to the box in the **GROUP BY** tab. The Designer adds the column to the bottom of the list.

The first column listed is used for primary grouping, the second column listed is used for secondary grouping, and so forth. To change the groupings, use one of the following options:

- Right-click the column and select *Move Up* or *Move Down*.
- Select the column and click the down or up arrow in the top right corner of the GROUP BY tab.

To remove a column, use one of the following options:

- Right-click the column and select *Delete*.
- Select the column and click the delete icon in the top right corner of the GROUP BY tab.

To group by complex expressions (instead of by specific column values), use another query to produce a single column containing the grouping expression. Insert the new query immediately before this transform in your data flow, and specify the created column on the GROUP BY tab.

If you specify a group by list, then all columns in the output schema must be either in the group by list or mapped to an aggregate function, such as avg, count, max, min, or sum.

This tab is similar to the GROUP BY clause in a SQL SELECT statement.

8.6.6.8.6 ORDER BY tab

Use the ORDER BY tab to specify the columns you want used to sort the output data set. To add a column, select the column in the input schema area and drag it to the box on the ORDER BY tab. The Designer adds the column to the bottom of the list.

The first column listed is used for primary sorting, the second column listed is used for secondary sorting, and so forth. To change the column order, use one of the following options:

- Right-click the column and select *Move Up* or *Move Down*.
- Select the column and click the down or up arrow in the top right corner of the **ORDER BY** tab.

To remove a column, use one of the following options:

- Right-click the column and select [Delete](#).
- Select the column and click the delete icon in the top right corner of the [ORDER BY](#) tab.

The default sort order is ascending. To change the order, select [Ascending](#) or [Descending](#) from the adjacent drop down box.

8.6.6.8.7 Advanced tab

Use the Advanced tab to set up Data Services so that it creates separate sub data flows. Sub data flows process any of the following resource-intensive query clauses:

- DISTINCT
- GROUP BY
- JOIN
- ORDER BY

Related Information

[Smart editor \[page 299\]](#)

[Functions and Procedures \[page 1035\]](#)

[pushdown_sql \[page 1181\]](#)

8.6.6.8.8 Find tab

Use the Find tab to search for a specific word or term in the input schema or the output schema.

Related Information

[Searching in an input or output schema \[page 945\]](#)

8.6.6.9 Joins in the Query transform

You can use the Query editor to define joins involving two or more tables. Specifying information on the FROM and WHERE tabs has the effect of creating FROM and WHERE clauses in a SQL SELECT statement. Supported join types are inner join, left outer join, and cross-product.

Begin by specifying join pairs and join conditions in the FROM tab. As needed, restrict the result set in either the FROM or WHERE tab depending on the information that you need the query to return.

i Note

The best practice is to define all joins in the FROM tab. However, inner joins can be defined in the WHERE tab using a WHERE clause.

For each pair of sources, the generated join proposal includes a join condition based on column names, foreign keys, or primary keys:

- **Foreign key:** If a foreign key relationship exists, Data Services adds a join condition to the expression for the columns related through keys. For example, if foreign key K2 of table T2 references primary key K1 of table T1, Data Services includes the join condition: `T1.K1=T2.K2`
- **Primary key and column name:** If a foreign key relationship does not exist, Data Services adds a join condition to the expression for columns with the same name where at least one column is part of a primary key. For example, suppose there is no foreign key relationship between tables T and S; however, both tables contain column A. Column A is part of the primary key in table S. In this example, Data Services includes the join condition: `T.A = S.A`

If neither foreign key nor primary key is satisfied, Data Services proposes no join condition for that pair of sources.

8.6.6.9.1 About join pair lists

The FROM tab allows you to create a SQL FROM clause by specifying join pair information including the tables to join, join type, and join condition. Join pairs are subject to the following requirements:

- In order to be used in a join pair, a table must be an input schema to the query.
- Inner joins and left outer joins can be specified in the same FROM clause.
- Any table from the schema list can be used as a source in at most one join pair. If a table is not explicitly joined to another table, then it will be cross-joined (Cartesian product) to the result of the final join defined by the join pairs list. A cross join (Cartesian product) is a special case of an inner join with an ON condition that always evaluates to TRUE. In other words, a cross join matches every row of one table with every row of another table.
- The join pairs list may be empty in the following cases:
 - the input schema for the query contains only one source.
 - each of the input schemas are intended to be cross-joined.
 - inner joins are defined using a WHERE clause in the WHERE tab.
- Identify a left source only for the first pair in the [Join pairs](#) list. All subsequent join pairs take the results of the preceding join pair as the left source.

For example, in the following screenshot, the first join pair is a left outer join with ODS_SALESORDER as the left source and ODS_CUSTOMER as the right source. The result of that join becomes the left source of the second join pair which is a left outer join with ODS_SALESITEM as the right source. Finally the result of the two left joins becomes the left source of an inner join with ODS_MATERIAL as the right source.

Mapping	SELECT	FROM	WHERE	GROUP BY	ORDER BY	Advanced	Find																									
<table border="1"> <thead> <tr> <th>Input schema(s)</th> <th>From</th> <th>Join rank</th> <th>Cache</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="checkbox"/> ODS_MATERIAL</td> <td><input checked="" type="checkbox"/></td> <td>0</td> <td>Yes</td> </tr> <tr> <td><input checked="" type="checkbox"/> ODS_SALESITEM</td> <td><input checked="" type="checkbox"/></td> <td>0</td> <td>Automatic</td> </tr> <tr> <td><input checked="" type="checkbox"/> ODS_SALESORDER</td> <td><input checked="" type="checkbox"/></td> <td>0</td> <td>Automatic</td> </tr> <tr> <td><input checked="" type="checkbox"/> ODS_CUSTOMER</td> <td><input checked="" type="checkbox"/></td> <td>0</td> <td>Automatic</td> </tr> </tbody> </table>								Input schema(s)	From	Join rank	Cache	<input checked="" type="checkbox"/> ODS_MATERIAL	<input checked="" type="checkbox"/>	0	Yes	<input checked="" type="checkbox"/> ODS_SALESITEM	<input checked="" type="checkbox"/>	0	Automatic	<input checked="" type="checkbox"/> ODS_SALESORDER	<input checked="" type="checkbox"/>	0	Automatic	<input checked="" type="checkbox"/> ODS_CUSTOMER	<input checked="" type="checkbox"/>	0	Automatic					
Input schema(s)	From	Join rank	Cache																													
<input checked="" type="checkbox"/> ODS_MATERIAL	<input checked="" type="checkbox"/>	0	Yes																													
<input checked="" type="checkbox"/> ODS_SALESITEM	<input checked="" type="checkbox"/>	0	Automatic																													
<input checked="" type="checkbox"/> ODS_SALESORDER	<input checked="" type="checkbox"/>	0	Automatic																													
<input checked="" type="checkbox"/> ODS_CUSTOMER	<input checked="" type="checkbox"/>	0	Automatic																													
<table border="1"> <thead> <tr> <th colspan="5">Join pairs:</th> </tr> <tr> <th>Left</th> <th>Join Type</th> <th>Right</th> <th></th> <th>Join Condition</th> </tr> </thead> <tbody> <tr> <td>ODS_SALESORDER</td> <td>Left outer join</td> <td>ODS_CUSTOMER</td> <td>...</td> <td>ODS_SALESORDER.CUST_ID = ODS_CUSTOMER.CUST_ID</td> </tr> <tr> <td>↳</td> <td>Left outer join</td> <td>ODS_SALESITEM</td> <td>...</td> <td>ODS_SALESITEM.SALES_ORDER_NUMBER = ODS_SALESORDER.SALES_ORDER_NUMBER</td> </tr> <tr> <td>↳</td> <td>Inner join</td> <td>ODS_MATERIAL</td> <td>...</td> <td>ODS_MATERIAL.MTRL_ID = ODS_SALESITEM.MTRL_ID</td> </tr> </tbody> </table>								Join pairs:					Left	Join Type	Right		Join Condition	ODS_SALESORDER	Left outer join	ODS_CUSTOMER	...	ODS_SALESORDER.CUST_ID = ODS_CUSTOMER.CUST_ID	↳	Left outer join	ODS_SALESITEM	...	ODS_SALESITEM.SALES_ORDER_NUMBER = ODS_SALESORDER.SALES_ORDER_NUMBER	↳	Inner join	ODS_MATERIAL	...	ODS_MATERIAL.MTRL_ID = ODS_SALESITEM.MTRL_ID
Join pairs:																																
Left	Join Type	Right		Join Condition																												
ODS_SALESORDER	Left outer join	ODS_CUSTOMER	...	ODS_SALESORDER.CUST_ID = ODS_CUSTOMER.CUST_ID																												
↳	Left outer join	ODS_SALESITEM	...	ODS_SALESITEM.SALES_ORDER_NUMBER = ODS_SALESORDER.SALES_ORDER_NUMBER																												
↳	Inner join	ODS_MATERIAL	...	ODS_MATERIAL.MTRL_ID = ODS_SALESITEM.MTRL_ID																												

- The join condition of a join pair cannot be empty and must refer to tables that are used in previous join pairs and the table used in the current join pair.
For example, the second join pair in the screenshot above may not refer to the table ODS_MATERIAL in its join condition because ODS_MATERIAL was not used in the previous join pair. The second join pair may only refer to ODS_SALESORDER, ODS_CUSTOMER, and ODS_SALESITEM.
- If the Query transform contains only inner joins, the WHERE tab may be used to specify join conditions. However, although valid, specifying join conditions in both the FROM tab and WHERE tab is not recommended. If conditions are specified in both the WHERE and FROM tabs, they are combined to form the join conditions for the query at job execution time.
- If a query contains a left outer join, any conditions specified in the WHERE tab are treated as filters. The join conditions for a left outer join may include multiple tables and are defined in the FROM tab.
- A Query transform in an ABAP data flow cannot contain mixed inner and left outer joins. The Query transform may have only inner joins, only left outer joins, or no joins at all.

8.6.6.9.2 Constructing a Join Query

You can use the Query editor to specify joins involving two or more tables. The resulting SQL join types may be inner join, left outer join, or cross product.

To construct a join query:


1. Within a data flow, connect the source tables to a Query transform.
2. Click the Query transform to open the Query editor.
3. Optionally, exclude input schemas by deselecting the *FROM* checkbox in uppermost area.

You may want to exclude an input schema if you no longer need its columns in the output schema, ORDER BY, and GROUP BY clauses.

It also may be useful to exclude nested input schemas. However, at least two input schemas must be selected in order to create a join.

4. In the lower pane of the Query editor, click the *FROM* tab.
5. Specify the join pairs and join conditions.

After the first join pair is specified, subsequent join pairs use the result of the previous pair as the left source. The following table describes the columns displayed in the *Join pairs* area.

Column name	Description
Left	<p>The left source of a join.</p> <p>For the first join pair, select the input schema from the drop-down list of available schemas.</p> <p>For subsequent join pairs, the result of the previous join pair is taken as the left source and the schema cannot be specified.</p>
Join type	<p>The type of join.</p> <p>Valid values are Inner join and Left outer join.</p> <p>If a table is not explicitly joined to another table, then it will be cross-joined (Cartesian Product) to the result of the final join defined by the join pairs.</p>
Right	<p>The right source of a join.</p> <p>Any input schema not used in a previous join pair.</p>
smart editor	Optional: click the icon to open the smart editor. Within the smart editor Data tab, you can drag and drop columns to specify the join condition.
...	
Propose Join	Optional: Click Propose Join to have Data Services generate a join expression.
	The SQL clause is automatically updated after you change the left source, right source, or join type.
Join condition	<p>A join condition is required for each join pair.</p> <p>Where possible, Designer automatically suggests a join condition based on the input schemas of the join pair. To enter or edit the join condition, you can type in the join condition field or use the smart editor.</p>

If your FROM clause contains only left outer joins or a mix of left outer joins and inner joins, you may want to change the order that the software executes the join pairs. Note that changing the execution order of the join pairs changes the results.

If your FROM clause contains only inner joins, changing the execution order does not change the results.

- As necessary, to filter the result set of a left outer join, place a restriction in the ON clause or within the WHERE tab.
- Optionally, specify the join ranks for each table in the [Join rank](#) column.

The join rank indicates the rank of the source relative to other tables and files in the data flow. The join rank has no effect on the actual result produced, but can have a profound effect on join performance. The software joins sources with higher join ranks before it joins sources with lower join ranks. The order of execution depends on join rank and, for left outer joins, the order defined in the FROM clause.

The join rank must be a non-negative integer. When set to its default value of 0, the software determines the join order. The join rank specified in the Query editor overrides any join rank specified in a source. For new jobs, specify the join rank in the Query editor.
- As necessary, specify desired caching in the [Cache](#) column.

Cache indicates whether the software should read the data from the source and load it into memory or pageable cache. Caching a source increases performance only when the data source is used as the right source of a join.

The cache value in the Query transform takes precedence over the value specified in a source. For newly created data flows, it is preferable to specify the cache value in the Query transform.

The default value in the Query transform is *Automatic*. Automatic assumes the value specified in the source.

i Note

The cache type, either in-memory or pageable, is set at the data flow level.

Related Information

[Restricting left outer joins \[page 966\]](#)

8.6.6.9.3 Join Examples

8.6.6.9.3.1 Joins using two sources

About inner and left outer joins

Sources joined in a Query transform produce different results based on the join type.

When joining two sources, an inner join returns rows from both sources where a match is found.

A left outer join returns the rows that meet the join condition, plus all of the rows in the left source that did not meet the join condition. Therefore, all rows in the left source are reproduced at least once in the result. Only data from the right source that satisfies the join condition appears in the result.

For rows from the left table that do not have corresponding data from the inner table, the missing values are assigned as null. In an inner join between the tables, the same rows would be absent in the result. The following diagram shows the difference in the join results for the sample data in the illustration:

Sources joined in a query produce different results based on the join type

Sources

Bonus Table

EmpID	Bonus
1008	1000
1009	1500
1011	1000

Employee Table

EmpID	LastName
1008	Alvarez
1009	Davis
1010	Tanaka
1011	Laprais

Join Condition

Employee.EmpID = Bonus.EmpID

Join Results

Inner Join

EmpID	LastName	Bonus
1008	Alvarez	1000
1009	Davis	1500
1011	Laprais	1000

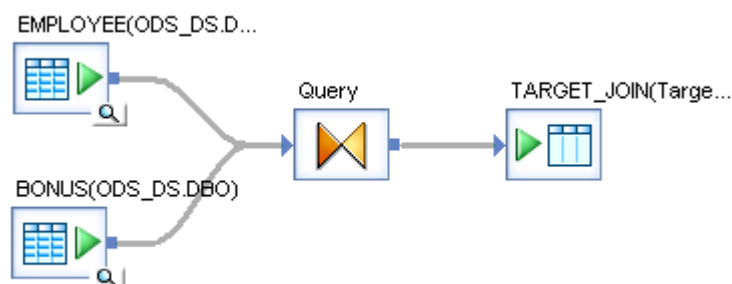
An inner join
produces 3 rows

Left Outer Join

EmpID	LastName	Bonus
1008	Alvarez	1000
1009	Davis	1500
1010	Tanaka	NULL
1011	Laprais	1000

A left outer join produces 4
rows where the additional
row is filled with NULLs

In Data Services, the two tables in the data flow would be sources to a Query transform and the join results would appear in the target table as shown below.



Inner join example

To produce a list of only the employees receiving bonuses, the data in the Employee Table and Bonus Table sources below are joined using an inner join to produce three rows where each row contains values from both sources.

The outer table is `Employee` and the inner table is `Bonus`.

Sources

Bonus Table

EmpID	Bonus
1008	1000
1009	1500
1011	1000

Employee Table

EmpID	LastName
1008	Alvarez
1009	Davis
1010	Tanaka
1011	Laprais

Join Results

Inner Join

EmpID	LastName	Bonus
1008	Alvarez	1000
1009	Davis	1500
1011	Laprais	1000

An inner join produces 3 rows

The syntax of the SQL is:

```
SELECT      EMPLOYEE.EMPID, EMPLOYEE.LASTNAME,
BONUS.BONUS
FROM EMPLOYEE INNER JOIN BONUS
ON          (EMPLOYEE.EMPID = BONUS.EMPID)
```

Use the Query editor [FROM](#) tab to define the Join pairs list as follows:

Left	Employee
Join Type	Inner join
Right	Bonus
Join Condition	EMPLOYEE.EMPID = BONUS.EMPID

Left outer join example

To produce a list of all employees and show which are receiving bonuses the data in the Employee Table and Bonus Table sources below are joined with a left outer join to produce four rows where the extra row contains a NULL.

The outer table is `Employee` and the inner table is `Bonus`.

Sources		
Bonus Table		Employee Table
EmpID	Bonus	EmpID LastName
1008	1000	1008 Alvarez
1009	1500	1009 Davis
1011	1000	1010 Tanaka
		1011 Laprais

Join Results		
Left Outer Join		
EmpID	LastName	Bonus
1008	Alvarez	1000
1009	Davis	1500
1010	Tanaka	NULL
1011	Laprais	1000

A left outer join produces 4 rows where the additional row is filled with NULLs

The SQL query is:

```
SELECT      EMPLOYEE.EMPID, EMPLOYEE.LASTNAME,
BONUS.BONUS
FROM EMPLOYEE LEFT OUTER JOIN BONUS
ON      (EMPLOYEE.EMPID = BONUS.EMPID)
```

Use the Query editor [FROM](#) tab to define the Join pairs list as follows:

Left	Employee
Join Type	Left outer join
Right	Bonus
Join Condition	EMPLOYEE.EMPID = BONUS.EMPID

8.6.6.9.3.2 Mixed inner and left outer joins

When joining more than two tables in the Query editor, a left source is identified in the first pair of tables in the [Join pairs](#) list. All subsequent join pairs take the results of the preceding join pair as the left source.

Mixed joins example

The example below illustrates how sequential joins can produce a result showing all of the departments that have employees and the employees' bonuses. In this case, the Department table would be the left source and the Employee table would be the right source of an inner join. The results of that inner join would then be joined to the Bonus table using a left outer join. The results of the inner join would be the left source and the Bonus table would be the right source.

The joins produce four rows with the bonus information NULL where there was no value in the bonus table.

Sources:

Department table		Employee table			Bonus table	
DeptID	Department	EmpID	LastName	DeptID	EmpID	Bonus
01	Accounting	1008	Alvarez	01	1008	1000
02	Finance	1009	Davis	02	1009	1500
03	Sales	1010	Tanaka	01	1011	1000
04	Marketing	1011	Laprais	01		

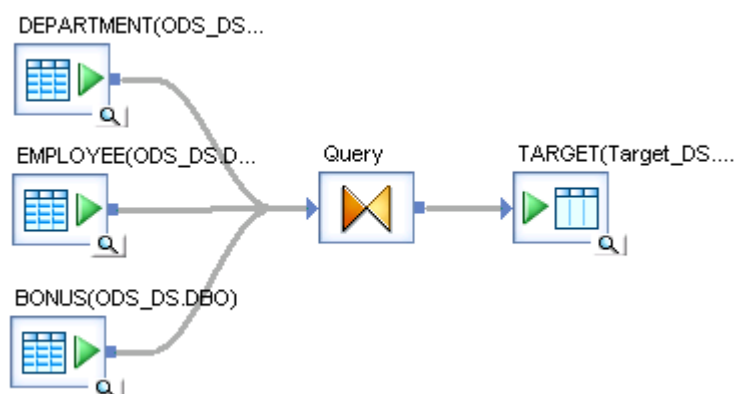
Join Results:

DeptID	Department	LastName	Bonus
1	Accounting	Alvarez	1000
1	Accounting	Tanaka	NULL
1	Accounting	Laprais	1000
2	Finance	Davis	1500

The SQL query is:

```
SELECT      DEPARTMENT.DEPTID, DEPARTMENT.DEPARTMENT,
            EMPLOYEE.LASTNAME, BONUS.BONUS
FROM (DEPARTMENT INNER JOIN EMPLOYEE
      (ON DEPARTMENT.DEPTID=EMPLOYEE.DEPTID) )
LEFT OUTER JOIN BONUS
ON      (EMPLOYEE.EMPID = BONUS.EMPID)
```

In Data Services, the three tables in the data flow would be sources to a Query transform and the join results would appear in the target table as shown below.



Use the Query editor *FROM* tab to define the Join pairs list as follows:

Left	Join Type	Right	Join Condition
DEPARTMENT	Inner join	EMPLOYEE	DEPARTMENT . DEPTID=EMPLOYEE . DEPTID
—»	Left outer join	BONUS	EMPLOYEE . EMPID=BONUS . EMPID

8.6.6.9.3.3 Restricting left outer joins

About restricting left outer joins

An unrestricted left outer join between two tables results in all of the rows from the left outer table along with data from the inner table that satisfies the join condition. NULL values are assigned in rows that do not contain data.

The result set for a left outer join that is restricted depends on whether the restriction is placed in the ON condition or in the WHERE clause. Where you place the restriction ultimately depends on what information you need the query to return. The following table shows the result set defined in each case and how to define the query in the Query editor.

Result set	Restriction	Query editor tab to use
Joined table includes all rows of the left source (including null values).	ON	FROM
Joined table includes only rows for which the restriction is true.	WHERE	WHERE

i Note

For inner joins, it does not matter where the restriction is placed; the result set is the same in either case.


Restriction placed in the ON condition example

For example, assume CUSTOMER is the left source and SALESORDER is the right source of a left outer join. The diagram below shows data in the source tables and the results of a left outer with the restriction placed in the ON condition.

Sources		
Customer Table		Sales Order Table
CUST_ID	Name	CUST_ID Order Number
DT03	Pottery Ceramics	SA01 PT22221000
KT02	Major Resellers	SA01 PT22221001
SA01	Trusty Manufacturers	SA01 PT22221002
SA02	New Times	SA02 PT22221005
SA03	Popular Press	SA02 PT22221006
		KT02 PT22221009
		KT02 PT22221010
		DT03 PT22221012

Join Results		
Restriction in ON condition		
CUST_ID	Name	Order Number
DT03	Pottery Ceramics	<Null>
KT02	Major Resellers	<Null>
SA01	Trusty Manufacturers	PT22221000
SA01	Trusty Manufacturers	PT22221001
SA01	Trusty Manufacturers	PT22221002
SA02	New Times	<Null>
SA03	Popular Press	<Null>

The join condition specified in the FROM clause contains the clause `AND CUSTOMER.CUST_ID = 'SA01'` and the result returns all rows of the left source, CUSTOMER, including those rows with NULL values. The Join pairs area of the FROM tab would appear as follows:

Join pairs:				
Left	Join Type	Right		Join Condition
ODS_CUSTOMER	Left outer join	ODS_SALESORDER	...	 ODS_SALESORDER.CUST_ID = ODS_CUSTOMER.CUST_ID and ODS_CUSTOMER.CUST_ID = 'SA01'

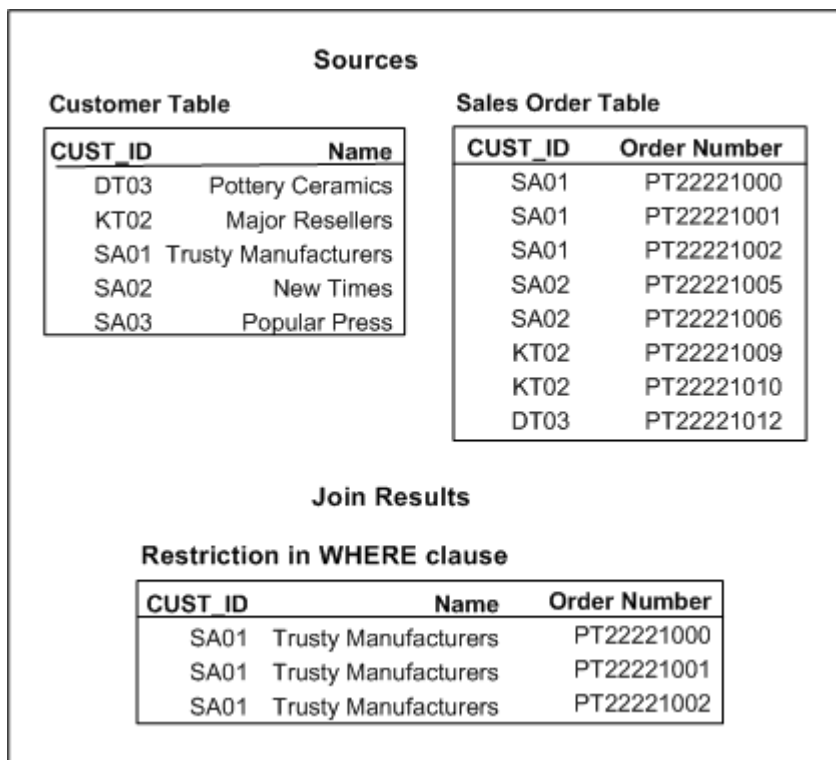
The SQL query is:

```
SELECT ODS_CUSTOMER.CUST_ID, ODS_CUSTOMER.NAME1,
       ODS_SALESORDER.SALES_ORDER_NUMBER
FROM DBO.ODS_CUSTOMER ODS_CUSTOMER LEFT OUTER JOIN
     DBO.ODS_SALESORDER ODS_SALESORDER
```

```
ON (ODS_SALESORDER.CUST_ID=ODS_CUSTOMER.CUST_ID)
AND (ODS_CUSTOMER.CUST_ID = 'SA01')
```

Restriction placed in the WHERE clause example

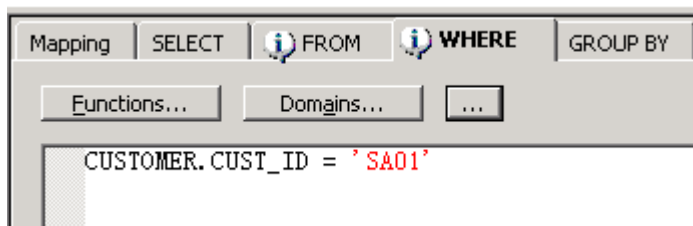
Referring again to the example of CUSTOMER as the left source and SALESORDER as the right source of a left outer join, if the restriction `CUSTOMER.CUST_ID = 'SA01'` is placed in the WHERE clause, the result of a left outer join returns only the rows for which the restriction is true. The data in the source tables and results of the join are shown in the diagram below.



The Join pairs area of the FROM tab would appear as follows:

Join pairs:				
Left	Join Type	Right		Join Condition
ODS_CUSTOMER	Left outer join	ODS_SALESORDER	...	ODS_SALESORDER.CUST_ID = ODS_CUSTOMER.CUST_ID

and the WHERE tab would contain the restriction:



The SQL query is:

```
SELECT ODS_CUSTOMER.CUST_ID, ODS_CUSTOMER.NAME1,
       ODS_SALESORDER.SALES_ORDER_NUMBER
FROM DBO.ODS_CUSTOMER ODS_CUSTOMER LEFT OUTER JOIN
     DBO.ODS_SALESORDER ODS_SALESORDER
ON (ODS_SALESORDER.CUST_ID=ODS_CUSTOMER.CUST_ID)
WHERE (ODS_CUSTOMER.CUST_ID = 'SA01')
```

8.6.6.9.4 Viewing Optimized SQL

Before running a job, you can view the SQL code that SAP Data Services generates for table sources in data flows. By examining the SQL code, you can verify that the software generates the commands you expect. If necessary, you can alter your design to improve the data flow.

To view the SQL code:

1. Validate and save data flows.
2. Open a data flow in the workspace.
3. Select *Display Optimized SQL* from the *Validation* menu.

Alternately, you can right-click a data flow in the object library and select *Display Optimized SQL*.

The *Optimized SQL* window opens and shows a list of datastores and the optimized SQL code for the selected datastore. By default, the *Optimized SQL* window selects the first datastore.

The software only shows the SELECT generated for table sources and INSERT INTO... SELECT for targets. It does not show the SQL generated for SQL sources that are not table sources, such as:

- Lookup function
 - Key_generation function
 - Key_Generation transform
 - Table_Comparison transform
4. Select a name from the list of datastores on the left to view the SQL that this data flow applies against the corresponding database or application.

i Note

The *Optimized SQL* window displays the existing SQL statement in the repository. If you changed your data flow, save it so that the *Optimized SQL* window displays your current SQL statement.

8.6.6.9.5 Outer join compared to the Lookup function

You can produce a similar data set using the lookup function. However, the lookup function is limited to the following actions:

Lookup function	Left Outer join
Returns only one column value for each comparison	Returns all column values for each comparison
Can be used against only one source at a time	Provides similar capability to multiple lookup calls
Cannot be used to produce the same results as non-equality joins (for example, $A.x < B.y$)	Allows non-equality joins
Permits default values other than nulls	
Can be cached when desired	

In addition, sources in an outer join query must be joined in a hierarchical order:

- A source can only be the inner table of one outer join
- A source cannot be "outer joined" with itself in a single Query transform

8.6.7 Row_Generation



Produces a data set with a single column. The column values start with the number that you set in the [Row number starts at](#) option. The value then increments by one to a specified number of rows.

Parent topic: [Platform transforms \[page 833\]](#)

Related Information

[Case \[page 833\]](#)

[Data Mask \[page 836\]](#)

[DQM Microservices \[page 925\]](#)

[Map_Operation \[page 938\]](#)

[Merge \[page 942\]](#)

[Query \[page 943\]](#)

[SQL \[page 972\]](#)

[User-Defined \[page 975\]](#)

[Validation \[page 983\]](#)

8.6.7.1 Data inputs

None.

8.6.7.2 Editor

The Row_Generation transform editor includes the target schema, and transform options.

8.6.7.3 Options

Option	Description
<i>Cache</i>	<p>Indicates whether the software should read the required data from the source and load it into memory or pageable cache. Because an inner source of a join must be read for each row of an outer source, you might want to cache a source when it is used as an inner source in a join.</p> <p>Options are:</p> <ul style="list-style-type: none"> • <i>Yes</i>: The source is always cached unless it is the outer-most source in a join. • <i>No</i>: The source is never cached. <p>The default is <i>Yes</i>.</p> <p>Cache specified in the Query transform editor FROM tab overrides any cache specified in a source. For new jobs, specify the cache only in the Query transform editor.</p>
<i>Join rank</i>	<p>Indicates the rank of the output data set relative to other tables and files joined in a data flow. The software joins sources with higher join ranks before joining sources with lower join ranks.</p> <p>Join rank specified in the Query transform editor FROM tab overrides any join rank specified in a source. For new jobs, specify the join rank only in the Query transform editor.</p> <p>Must be a non-negative integer. Default value is 0.</p>
<i>Row count</i>	<p>A positive integer indicating the number of rows in the output data set. For added flexibility, you can enter a global variable or substitution parameter.</p>

Option	Description
<i>Row number starts at</i>	An integer with which row numbering starts. If you set this option to 1, the first row will be labeled 1, the second row will be labeled 2, and so on. If you leave this blank, numbering will start at zero (0). For added flexibility, you can enter a global variable or substitution parameter.

8.6.7.4 Data outputs

The Row_Generation transform produces a data set with a single column and the number of rows specified in the *Row count* option. The rows contain integer values in sequence starting from the value that you entered in the *Row number starts at* option, and incrementing by one in each row.

8.6.8 SQL



Performs the indicated SQL query operation.

Use this transform to perform standard SQL operations when other built-in transforms cannot perform them.

The options for the SQL transform include specifying a datastore, join rank, cache, array fetch size, and entering SQL text.

i Note

The SQL transform supports a single `SELECT` statement only.

Parent topic: [Platform transforms \[page 833\]](#)

Related Information

[Case \[page 833\]](#)

[Data Mask \[page 836\]](#)

[DQM Microservices \[page 925\]](#)

[Map_Operation \[page 938\]](#)

[Merge \[page 942\]](#)

[Query \[page 943\]](#)

[Row_Generation \[page 970\]](#)

[User-Defined \[page 975\]](#)

[Validation \[page 983\]](#)

[XML_Map \[page 994\]](#)

8.6.8.1 Data inputs

None. This transform does not allow an input data set.

8.6.8.2 Options

Option	Description
Array fetch size	<p>Indicates the number of rows retrieved in a single request to a source database. The default value is 1000. Higher numbers reduce requests, lowering network traffic, and possibly improve performance. The maximum value is 5000.</p> <p>This option is available for source tables from DB2, Informix, ODBC, Oracle, and SQL Server datastores.</p> <p>When retrieving a column with an Oracle long data type, Data Services automatically sets Array Fetch Size to 1. If a column has an Oracle long data type, Data Services can only retrieve one row at a time.</p>
Cache	<p>Indicates whether the software should load output from the transform into memory or pageable cache. Because an inner source of a join must be read for each row of an outer source, you might want to cache output when it is used in a subsequent transform as an inner source in a join.</p> <p>Options are:</p> <ul style="list-style-type: none">• Yes: The source is cached.• No: The source is not cached. <p>The default is Yes.</p> <p>Cache specified in the Query transform editor FROM tab overrides any cache specified in a previous transform within the data flow. For new jobs, specify the cache only in the Query transform editor.</p>

Option	Description
<i>Database type</i>	<p>Database type and versions available in the selected datastore.</p> <p>The <i>Database type</i> box allows you to quickly set SQL transform values in data flows if you have multiple configurations in a datastore. This option also allows you to supply unique SQL text for each database type and version in any SQL transform instance. When you select a database type, Data Services provides the value you set previously for SQL text in that particular database type. To add or remove items in the <i>Database type</i> list box, edit the datastore configuration information using the Datastore Editor.</p> <p>The following describes how Data Services determines SQL text values. For more information about the Datastore Editor and its dialogs, see Datastore editor [page 57].</p> <ul style="list-style-type: none"> If the datastore has more than one configuration and there are different database types or versions, then Data Services determines the initial SQL text values for the additional database types and versions from the <i>Use values from</i> box in the Create New Configuration dialog box (a sub-dialog of the Datastore Editor). <div style="background-color: #f0f0f0; padding: 10px; margin: 10px 0;"> <p>i Note</p> <p>Join rank, Cache, and Array fetch size values remain the same as those set in the initial configuration. You cannot have more than one set of these values in a datastore.</p> </div> <ul style="list-style-type: none"> If you also select the <i>Restore values if they already exist</i> check box in the Create New Configuration dialog box, Data Services looks for previously defined values that once existed for that database type or version. It is possible for a data flow to contain SQL transform values for a database type or version, even if its datastore configuration was deleted. Data Services retains all SQL transform values saved with every datastore configuration. If such values exist, then Designer restores those values. Otherwise, it gets the values from the configuration you select from the <i>Use values from</i> option. <p>If the SQL text in a SQL transform is not correct for the database type, modify the SQL text. If the SQL text has any hard-coded owner names or database names in it, consider replacing them with variables to limit the number of modifications you need for new database types.</p> <div style="background-color: #f0f0f0; padding: 10px; margin: 10px 0;"> <p>i Note</p> <p>Because Data Services only provides values for variables during run time, do not use variables in the SQL text of a SQL transform when you use the <i>Update Schema</i> button. To support portability, add variables afterwards.</p> </div>
<i>Datastore</i>	The name of the datastore that Data Services uses to access the tables referred to in <i>SQL text</i> .
<i>Join rank</i>	<p>Indicates the rank of the output data set relative to other tables and files joined in a data flow. The software joins sources with higher join ranks before joining sources with lower join ranks.</p> <p>Join rank specified in the Query transform editor FROM tab overrides any join rank specified in a source. For new jobs, specify the join rank only in the Query transform editor.</p> <p>Must be a non-negative integer. Default value is 0.</p>

Option	Description
SQL text	<p>The text of the SQL query. This string is passed to the database server.</p> <p>You do not need to put enclosing quotes around the SQL text. You can put enclosing quotes around the table and column names, if required by the syntax rules of the DBMS involved.</p>
Update schema	Click this option to automatically calculate and populate the output schema for the SQL SELECT statement.

8.6.8.3 Data outputs

There are two ways of defining the output schema for a SQL transform:

- **Automatic:** After you type the SQL statement, click [Update schema](#) to execute a described select statement against the database which obtains column information returned by the select statement and populates the output schema.
- **Manual:** Output columns must be defined in the output portion of the SQL transform.

The number of columns defined in the output of the SQL transform must equal the number of columns returned by the SQL query.

The column names and data types of the output columns need not match the column names or data types in the SQL query. Data Services conversion rules apply.

When possible, Data Services optimizes data flows by pushing expressions down to an underlying database manager. In a single transaction, Data Services can push down expressions so that they are performed by the underlying database manager. However, when Data Services evaluates an expression which includes operands of more than one data type, Data Services attempts to convert the operands to the same data type first. (Except for national character-set data types which can be pushed down while others in an expression are not.). Errors are flagged for illegal conversion.

The output data set cannot contain hierarchical data.

Exercise care when specifying the output columns. Typically the column data types of the two sets of columns should be an exact match. If you choose to have different data types, you need to ensure that they are compatible—if they are not, you will get a runtime error from the underlying database manager.

Related Information

[varchar \[page 316\]](#)

8.6.9 User-Defined



The User-Defined transform provides you with custom processing in a data flow using full Python scripting language. The applications for the User-Defined transform are nearly limitless. It can do just about anything that you can write Python code to do. You can use the User-Defined transform to generate new records, populate a field with a specific value, create a file, connect to a website, or send an email, just to name a few possibilities.

You can place this transform anywhere in your data flow. If you have created your own transform, then the only restrictions about where it can be located in the data flow are those which you place on it.

Although the User-Defined transform is quite flexible and powerful, you will find that many of the tasks you want to perform can be accomplished with the Query transform. The Query transform is generally more scalable and faster, and uses less memory than User-Defined transforms.

Editors

Similar to Data Quality transforms, the User-Defined transform has a transform editor which contains the Input, Options, and Output tabs.

Unlike most Data Quality transforms, however, you cannot edit options in the User-Defined transform editor. To edit options, you must use the User-Defined editor, which is accessed from the Options tab in the User-Defined transform editor or from the Tools menu.

You may also notice some options displayed in the Options tab of the User-Defined transform editor that are not displayed in the User-Defined editor. These options are not editable.

Caution

Make sure that if you use an input field in a Python expression in your User-Defined transform, you first map it to a recognized field name in the Input tab. If it is not mapped, you will receive an error message similar to the following:

```
def GetField(*args): return apply(_flpythonmodulesu.FlDataRecord_GetField,args)
RuntimeError: FlDataRecord::GetField() error: Invalid field name
MAPPED_RECNO.
```

Parent topic: [Platform transforms \[page 833\]](#)

Related Information

[Case \[page 833\]](#)

[Data Mask \[page 836\]](#)

[DQM Microservices \[page 925\]](#)

[Map_Operation \[page 938\]](#)

[Merge \[page 942\]](#)

[Query \[page 943\]](#)

[Row_Generation \[page 970\]](#)

[SQL \[page 972\]](#)

[Validation \[page 983\]](#)

[XML_Map \[page 994\]](#)

8.6.9.1 Content objects

We provide content objects to help you create and complete transforms.

Transform configurations

A transform configuration is a transform with preconfigured input fields, output fields, and options that can be used in multiple data flows. These are useful if you repeatedly use a transform with specific options and input and output fields.

When Data Services is installed, read-only transform configurations are provided for the User-Defined transform.

You can use transform configurations in your data flows or as an example of a typical transform. After you place an instance of the transform configuration in a data flow, you can override these preset defaults. You can also create your own transform configuration, either by replicating an existing transform configuration or creating a new one.

Sample blueprints and other objects

We have created Data Quality blueprints and other content objects to help you set up Data Services jobs. We've identified a number of common scenarios that you are likely to perform with Data Services. For each scenario, we've included a blueprint that is already set up to solve the business problem in that scenario.

Related Information

[Downloading blueprints and other content objects \[page 444\]](#)

[Transform configurations \[page 440\]](#)

8.6.9.2 User-Defined options

The User-Defined transform contains options that determine how the transform processes data. Many User-Defined transform options are also found in the Match transform.

Option	Description
Mode	<p>Specifies how the Python expression is applied to the transform.</p> <p><i>Per collection:</i> Applies the expression to entire data collection. Use this option when adding new records, which did not exist before, into the data flow. Selecting this option displays the Group Forming option group, in which you set up break groups and candidate selection.</p> <p><i>Per record:</i> Applies the expression to each record. You cannot add new records into the data flow with this option. This option is the default and what you will want to use most often.</p>

Related Information

[Group forming \[page 978\]](#)

[User-Defined Transform options \[page 982\]](#)

8.6.9.2.1 Group forming

Group forming allows you to group and prioritize records for better match accuracy and efficiency, as well as performing custom Python processing.

Break groups

Break groups allow you to group records based on common field values.

Use break groups to lower the number of comparison the Match transform needs to make and to increase the speed of the matching process.

Candidate selection

The process of candidate selection appends records from a relational database to an existing data collection for processing.

For real-time jobs, candidate selection pulls a candidate set of records based on a single record or many records.

To display the Candidate Selection option group, right-click the Group Forming option group and select Add Candidate Selection.

i Note

Candidate selection works with relational databases only; it does not work with flat files.

Related Information

[Break group options \[page 658\]](#)

[Candidate selection options \[page 660\]](#)

8.6.9.2.1.1 Break group options

Use the break group options to group records based on common field values.

Option	Description
Split records into break groups	<p>Select this option if you want to form break groups to reduce the total number of comparisons made.</p> <p>The most common case for deselecting this option is when have a real-time job and your data comes in as one break group. This scenario also often makes use of candidate selection (selecting a limited number of records from a relational database) for optimal real-time matching.</p> <div><p>⚠ Caution</p><p>Deselect this option with caution within a batch data flow. The size of a break group may not exceed 2 GB. If you use this option in a batch data flow, also set the Maximum allowable break group size (in records) option so that the collection does not exceed the size limit. If it does exceed the limit, the data flow will abort.</p></div> <div><p>i Note</p><p>Break group size is calculated by multiplying the record length by the number of records in the break group.</p></div>
Field	<p>Choose a mapped input field name from the drop-down menu that you want to include in the break key. Click the Add Row button to add another field.</p> <p>If you require a more complex break key, you could define that field using an upstream Query transform and select the field here.</p>
Start Position	<p>Enter the start position of the field. Valid values for a field of n are 1 to n and -1 to -n. Negative start values signify that the start position is counted from the right.</p> <p>For example, a field with a length of 7 contains JOHNSON. A start position of 2 would mean start with "O." A start position of -4 means start with the "N" (This would also be the case if the field has a length of 20, because the negative start value starts from the actual length of the string, not of the field).</p>
Length	<p>Enter the number of characters in the field you want included in the break key.</p>

Option	Description
<i>Break key case sensitive</i>	<p>Specifies whether to treat the break key as case sensitive.</p> <p>Yes: Treat the break key as case sensitive.</p> <p>No: Do not treat the break key as case sensitive.</p> <p>For example, if you create a break key using the primary name (street), separate break groups would be formed with values of "Main" and "main" when you specify that the break key is case sensitive.</p>
<i>Replace NULL with empty string</i>	<p>Specifies whether to convert NULL values with an empty string in the break key.</p> <p>Yes: Convert NULL to an empty string.</p> <p>No: Do not convert to an empty string.</p>
<i>Right pad fields with blanks</i>	<p>Because the break key is used for sorting and aggregating, it is sensitive to the position in which data is placed. By right-padding the break key fields you can help ensure that break groups are formed properly.</p> <p>If the <i>Replace NULL with empty string</i> option is set to YES and this option is set to YES, then fields with NULL values will be replaced with all spaces (to the length of the field).</p> <p>Yes: Right-pad fields with blank spaces.</p> <p>No: Do not right-pad fields.</p>
<i>Input already sorted</i>	<p>Specifies that the input data has already been sorted, and you do not want it sorted again.</p> <p>For example, if you require a more complex break key, you could use a Query transform to create it, and use the ORDER BY operation to order your data.</p> <p>Yes: The transform will not re-sort the input data.</p> <p>No: The transform will sort the break keys at runtime before forming break groups.</p>
<i>Maximum allowable break group size (in records)</i>	<p>Specifies the maximum number of records allowed in a break group. An empty value or zero means that there is no limit on the break group size.</p> <p>With this option, you can control the amount of memory used during processing by specifying the number of records processed at one time.</p> <p>If more records make it into a single break group than specified, then the data flow throws an error and stops.</p>

8.6.9.2.1.2 Candidate selection options

The candidate selection option group includes the following options:

Option	Description
<i>Datastore</i>	<p>Select a valid datastore.</p> <p>This list is populated with all valid SQL and persistent cache datastores.</p> <p>If you choose a persistent cache datastore, you will not be able to enter custom SQL.</p>
<i>Cache type</i>	<p>This option can be used to improve performance, with a trade-off of more memory consumption.</p> <p><i>No_Cache</i>: Specifies that each query will be sent to the database.</p> <p><i>Pre_Load_Cache</i>: Specifies that the entire secondary table is cached to a local disk or memory.</p>
<i>Auto-generate SQL</i>	Select to have your SQL generated by the transform. This option allows you to query a simple single table. If you need to join tables or create a complex WHERE clause, you should select the <i>Create custom SQL</i> option.
<i>Table</i>	Enter a valid table name from the datastore.
<i>Use break column from database</i>	Select this option if your database already contains a column that corresponds to the break key field.
<i>Break key field</i>	Select the column from the secondary table that contains the break key field.
<i>Create custom SQL</i>	Select to create custom SQL.
<i>Launch SQL Editor</i>	Opens the SQL editor. This button is only enabled if you select the <i>Create custom SQL</i> option.
<i>Use constant source value</i>	Select to assign records to a physical source for generating appropriate statistics.
<i>Physical source value</i>	Type a value for your physical source. This value will be placed in the physical source field you select.
<i>Physical source field</i>	Select the mapped field that contains the physical source name.
<i>Add DB columns to mapping table</i>	<p>If you are using the <i>Create custom SQL</i> option, clicking this button will add only the database columns that appear in the SELECT statement and in the order that they appear in the SELECT statement.</p> <p>If you are using the <i>Auto-generate SQL</i> option, clicking this button will add ALL database columns, in the order that they appear in the table schema.</p>

i Note

If you do not associate an input field to any of these columns in the column mapping table, they will be removed when you close the window.

Column mapping table

This table allows you to specify which mapped field in the data flow each database selected field is assigned to.

Column	Description
<i>Break key</i>	Specifies whether this field is used as part of your break key.
<i>Field</i>	Each cell contains a list of the mapped names from the input fields in the transform.
<i>DB column</i>	Each cell contains a list of the column names in your database table or the selected columns from a custom query. Match the data of a column in your database to the data of a mapped field.

8.6.9.2.2 User-Defined Transform options

Custom options table

This table allows you to create custom options to be used as variables in your Python expression. The custom options are only available within the User-Defined transform. These options adjust the User-Defined transform's run-time behavior. Add or remove rows by using the buttons.

Option	Description
Custom option	Specifies variables for use in your Python expression.
Value	Specifies the value of the custom option.

The User-Defined Transform option group also contains the following option:

Option	Description
Run as separate process	<p>This option creates a separate sub data flow process for the transform when Data Services executes the data flow.</p> <p><i>Yes:</i> Splits transform into separate process.</p> <p><i>No:</i> Keeps transform in same process as the rest of the data flow.</p>

8.6.9.2.2.1 Python Expression Editor option

This option group contains the actual Python expression that the User-Defined transform will use. This option group is required.

Click the Launch Python Editor button to access the Python Expression editor.

Option	Description
Python	Displays the Python expression that will be applied to the transform. You can enter the Python code here or use the Python Expression editor.

Caution

Make sure to use a "u" to indicate Unicode every time you use a Unicode string to look up field names; for example, in a GetField, SetField, or SendToPipe method. If you do not, an error or crash may occur.

Related Information

[Creating an expression with the Python Expression editor \[page 1403\]](#)

[Python \[page 1400\]](#)

8.6.10 Validation



The Validation transform qualifies a data set based on rules for input schema columns. You can apply multiple rules per column or bind a single reusable rule (in the form of a validation function) to multiple columns.

The Validation transform can identify the row, column, or columns for each validation failure. You can also use the Validation transform to filter or replace (substitute) data that fails your criteria.

When you enable a validation rule for a column, a check mark appears next to it in the input schema.

Parent topic: [Platform transforms \[page 833\]](#)

Related Information

[Case \[page 833\]](#)

[Data Mask \[page 836\]](#)

[DQM Microservices \[page 925\]](#)

[Map_Operation \[page 938\]](#)

[Merge \[page 942\]](#)

[Query \[page 943\]](#)

[Row_Generation \[page 970\]](#)

[SQL \[page 972\]](#)

[User-Defined \[page 975\]](#)

[XML_Map \[page 994\]](#)

8.6.10.1 Validation Rules tab options

Open the Validation transform editor by clicking the name of the transform in your data flow. On the Validation Rules tab, the top pane lists all of the available rules and the bottom pane lets you define substitution values for failed rules to send to the Pass output schema.

The following table describes the options for the Validation Rules tab.

Type	Option	Description
Rules buttons	Add	Click to add a new rule. Launches the Rule Editor dialog box.
	Edit	Select an existing rule and click to edit the rule. You can also double-click a rule to open the Rule Editor.
	Remove	Select one or more rows and click to delete the rule(s).
Rules columns	Enabled	Select to enable the rule; clear to disable it. You can also enable or disable a rule in the Rule Editor dialog box.
	Rule	The syntactical name of the rule.
	Ignore if NULL	If set to Yes, Data Services skips (ignores) the rule if any of the associated column values are NULL. You set this option in the Rule Editor.
	Action on Fail	Identifies what action to take when the row fails: send the row to the Fail target, Pass target, or both. If you choose Send to Pass or Send to Both , you can choose to substitute a value or expression for the failed values that go to the Pass output using the bottom pane of the Validation Rules tab.
	Name	The common name as defined in the Name field of the Rule Editor.
If rule(s) fails and send to Pass, substitute:	Description	An optional description as defined in the Rule Editor.
	Remove button	Select one or more rows and click to delete the substitution(s).
	Enabled	Select to enable the substitution; clear to disable it.
	Column	Identifies the column to which the substitution will apply. Double-click the cell to enable the drop-down list.

Type	Option	Description
	Expression	Defines the substitution constant, variable, or function call. Double-click the cell to type in the cell. During job execution, Data Services converts substitute values to a corresponding column data type: integer, decimal, varchar, date, datetime, timestamp, or time.
	Ellipses button	As an alternative to typing a value in the Expression cell, click the ellipses button to launch the Smart Editor, where you can configure variables, substitution parameters, and functions for defining the substitution.

Related Information

[Validation Transform Options tab \[page 985\]](#)

8.6.10.2 Validation Transform Options tab

The Validation Transform Options tab has the following options:

Option	Description	
On failure:	Collect data validation statistics	Select this option to generate statistics for columns that failed validation to view in the Data Validation dashboards metadata reports.
	Collect sample data	Select this option to capture sample data for columns that failed validation to view in Data Validation dashboard metadata reports.
Output Rule Violation Information	Create column DI_ROWID on Validation_Fail	Select to include a DI_ROWID column in the Fail output schema (selected by default).

Related Information

[Validation Rules tab options \[page 984\]](#)

8.6.10.3 Rule Editor

You use the Rule Editor to define or edit a validation rule. Launch the Rule Editor from the Validation Rules tab of the Validation transform by clicking [Add](#) or double-clicking an existing rule to edit it.

The Rule Editor lets you configure a rule either based on an existing validation function or a rule defined in the validation transform that is based on a single column input.

Rules defined with a validation function have the following characteristics:

- Reusable
- Accept multiple input parameters
- Preferable for more complex rules
- Can be created in SAP Information Steward and imported and used in Data Services

Rules defined with a column validation have the following characteristics:

- Not reusable; the rule definition is part of the transform (not shared outside the transform)
- Binds to only one input column
- Better for simple rules

Also note there are two types of validation functions as categorized in the object library:

- Imported from Information Steward: These functions were created in Information Steward and cannot be edited in Data Services
- Locally created: Created and editable in Data Services

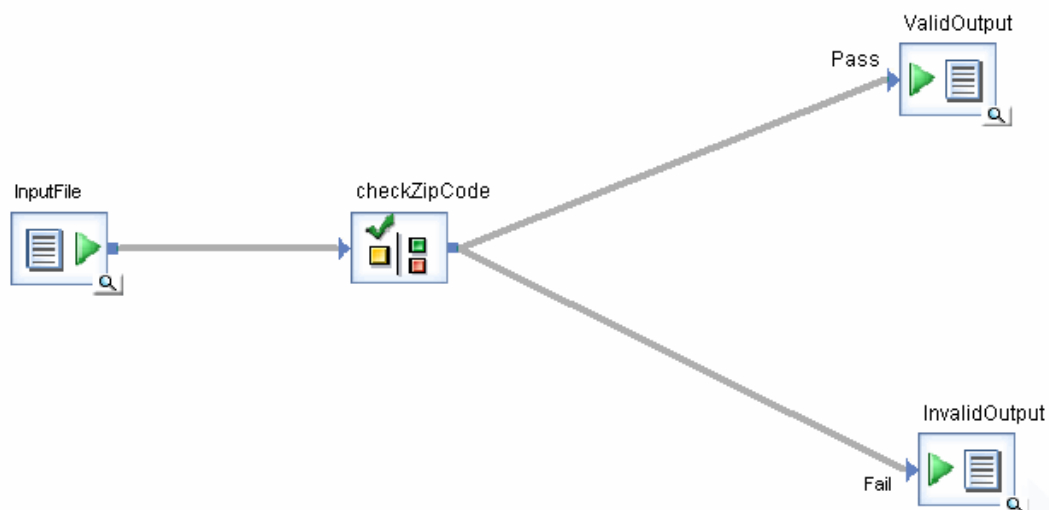
8.6.10.3.1 Defining a validation rule based on a column

1. In a data flow with the Validation transform connected to an input schema, click the transform name to open its editor.
2. On the Validation Rules tab, click [Add](#).
The Rule Editor displays with the [Enabled](#) check box selected by default.
3. Type a name for the rule and optionally add a description.
4. Select an [Action on fail](#):
 - Send To Fail
 - Send To Pass
 - Send To Both
5. Select [Column Validation](#).
6. Select a [Column](#) from the drop-down list.
7. Define a condition. All conditions must be Boolean expressions. Or, select [Custom Condition](#) to enable access to the smart editor (ellipses button) or function wizard ([Functions](#) button).

❖ Example

Defining a validation rule for a five-digit ZIP code pattern

The following example defines a Validation transform that verifies that the data for ZIP code is a five-digit number.



1. Open the Validation transform editor.
2. Click *Add*.
3. Name the rule **checkzip**.
4. For *Action on Fail*, select *Send To Both*.
5. Select *Column Validation*.
6. Select the *Customers.Zipcode* column.
7. Select the *LIKE* condition.
8. Specify the pattern as **99999**, which matches any five-digit number.
9. Click *OK*.
10. On the *Validation Rules* tab, for the option, *If any rule fails and send to Pass, substitute with:* double-click a cell under *Column* and select *Customers.Zipcode*.
11. Double-click in the cell under *Expression* and type '**Invalid**'.

If the zip code does not match the pattern, Data Services replaces the value with the string **Invalid**.

The following graphic shows the input data.

Q "C:\work\validation_data\demo\orderinfo2.txt"(OrderInfo)

	OrderNum	OrderDate	Name	Address1	Address2	City	State	Zipcode	Phone
▶	5001	2004.05.12	First National bank	1 First Street	NULL	San Jose	CA	95134	408-933-6821
	5002	NULL	Second National bank	123 First Street	NULL	San Jose	CA	95134	408-933-3456
	3500	2004.05.01	Second National bank	2 First Street	NULL	San Jose	CA	95134	408-933-6821
	5003	2004.05.12	First National bank	1 First Street	NULL	San Jose	CA	NULL	408-933-6821
	1	2004.06.12	First National bank	123 First Street	NULL	San Jose	CA	95134	408-933-6821
	5005	NULL	Third National bank	234 Commercial Street	NULL	San Francisco	CA	95100	650-933-3456
	5006	2004.05.16	First National bank	1 First Street	NULL	San Jose	CA	95134	408-933-6821
	5007	2004.05.16	First National bank	1 First Street	NULL	San Jose	CA	951ab	408-933-6821
	5008	2004.06.18	Third National bank	234 Commercial Street	NULL	San Francisco	CA	95100	650-933-3456
	5009	NULL	Third National bank	234 Commercial Street	NULL	San Francisco	CA	95100	650-933-3456
	5007	2004.05.16	First National bank	1 First Street	NULL	San Jose	CA	1234567	408-933-6821

The following graphic shows the data sent to the Pass/ValidOutput table:

Q "C:\work\validation_data\demo\orderinfo2_out.txt"(OrderInfo)

	OrderNum	OrderDate	Name	Address1	Address2	City	State	Zipcode	Phone
▶	5001	2004.05.12	First National bank	1 First Street	NULL	San Jose	CA	95134	408-933-6821
	5002	NULL	Second National bank	123 First Street	NULL	San Jose	CA	95134	408-933-3456
	3500	2004.05.01	Second National bank	2 First Street	NULL	San Jose	CA	95134	408-933-6821
	5003	2004.05.12	First National bank	1 First Street	NULL	San Jose	CA	Invalid	408-933-6821
	1	2004.06.12	First National bank	123 First Street	NULL	San Jose	CA	95134	408-933-6821
	5005	NULL	Third National bank	234 Commercial Street	NULL	San Francisco	CA	95100	650-933-3456
	5006	2004.05.16	First National bank	1 First Street	NULL	San Jose	CA	95134	408-933-6821
	5007	2004.05.16	First National bank	1 First Street	NULL	San Jose	CA	Invalid	408-933-6821
	5008	2004.06.18	Third National bank	234 Commercial Street	NULL	San Francisco	CA	95100	650-933-3456
	5009	NULL	Third National bank	234 Commercial Street	NULL	San Francisco	CA	95100	650-933-3456
	5007	2004.05.16	First National bank	1 First Street	NULL	San Jose	CA	Invalid	408-933-6821

The following graphic shows the data sent to the Fail/InvalidOutput table:

Q "C:\work\validation_data\demo\orderinfo2_failed.txt"(FailedOrderInfo)

	Ord...	OrderDate	Name	Address1	Addr...	City	State	Zipcode	Phone	D	DI_ERRORCOLUMNS
▶	5003	2004.05.12	First National bank	1 First Street	NULL	San Jose	CA	NULL	408-933-6821	B	checkZipCode failed rule(s): Zipcode
	5007	2004.05.16	First National bank	1 First Street	NULL	San Jose	CA	951ab	408-933-6821	B	checkZipCode failed rule(s): Zipcode
	5007	2004.05.16	First National bank	1 First Street	NULL	San Jose	CA	1234567	408-933-6821	B	checkZipCode failed rule(s): Zipcode

8.6.10.3.2 Defining a validation rule using a custom validation function

This section describes how to first add and define the new custom validation function in the object library, then how to use the function to define a validation rule.

8.6.10.3.2.1 Defining a custom validation function in the object library

This procedure describes how to add and define a custom validation function in the object library.

In this example, the function determines whether the ZIP column contains 5-digit ZIP codes by checking each character and ensuring that each character is a digit. To verify the character is a digit, the function checks to see if its ASCII value is between 48 and 57 inclusive (corresponding to 0 through 9 respectively).

1. From the *Custom Function* tab of the object library, right-click *Validation Functions* and select *New*. The *Custom Function* dialog box displays with *Validation function* selected.
2. Enter the name **zipIsValid** for the new function and click *Next*. The smart editor displays.
3. To define the parameters and variables for this function, select the *Variables* tab in the smart editor window.
4. Right-click the *Parameters* node and select *Insert*.
5. Enter a parameter name such as **\$ZipToValidate**.
6. Select the appropriate data type such as *varchar* with a length of **5**.
7. For the *Parameter type* select *Input*.

i Note

Validation functions can have multiple input parameters; additional output parameters are not used.

8. Click *OK*. The parameter \$ZipToValidate appears under the *Parameters* node.
9. Add five local variables named \$Z1 through \$Z5, which represent the five characters in the column. Right-click *Local*, select *Insert*, enter the name, select the data type *int*, and click *OK*. The five variables appear under the *Local* variables node.
10. In the text panel of the smart editor, enter the following validation script:

```
$Z1 = ascii(substr($ZipToValidate, 1, 1));
$Z2 = ascii(substr($ZipToValidate, 2, 1));
$Z3 = ascii(substr($ZipToValidate, 3, 1));
$Z4 = ascii(substr($ZipToValidate, 4, 1));
$Z5 = ascii(substr($ZipToValidate, 5, 1));
if ( $Z1 < 48 OR $Z1 > 57 )
  return 0;
if ( $Z2 < 48 OR $Z2 > 57 )
  return 0;
if ( $Z3 < 48 OR $Z3 > 57 )
  return 0;
if ( $Z4 < 48 OR $Z4 > 57 )
  return 0;
if ( $Z5 < 48 OR $Z5 > 57 )
  return 0;
return 1;
```

i Note

The Validation transform only supports custom functions that return an integer data type. If a return value is not a zero, then Data Services processes it as TRUE.

11. Click [OK](#).

The new function displays in the object library under the [Validation Function](#) node.

Continue to the next procedure to use the validation function in a validation rule.

8.6.10.3.2 Defining a validation rule using a validation function

This procedure describes how to use a custom validation function to define a validation rule.

1. Add a data flow with a source and a Validation transform and connect them.
2. Click the transform name to open the Validation transform editor.
3. On the [Validation Rules](#) tab, click [Add](#).
The Rule Editor displays with the [Enabled](#) check box selected by default.
4. In the Rule Editor, name the rule.
5. Select the [Action on fail: Send to Fail](#).
6. With the [Validation Function](#) option selected, from the drop-down list select the `ZipIsValid` function.
The Bindings table populates with the required parameters for the function.
7. Define the parameter argument. For the `$ZipToValidate` parameter, double-click the cell under [Argument](#) and select the column [ZIP](#).
8. Click [OK](#) to save the rule.
The rule appears in the Rules list on the [Validation Rules](#) tab.

When you run the job, the rule will apply the reusable validation function to the ZIP column and check each character to validate each is a digit.

8.6.10.3.3 Rule Editor options

The following table describes the options in the Rule Editor dialog box.

Option	Description
Name	The rule name as created by the user.
Description	An optional description provided by the user.
Enabled	Select to apply the rule when running the job. Clear to disable the rule. You can also enable or disable the rule on the Validation Rules tab.
Ignore if NULL	If set to Yes, Data Services skips (ignores) the rule if any of the associated column values are NULL.
Action on Fail	Identifies what action to take when the row fails: send the row to the Fail target, Pass target, or both. If you choose Send to Pass or Send to Both , you can choose to substitute a value or expression for the failed values that go to the Pass output in the bottom pane of the Validation Rules tab.

Option	Description
Validation Function	Select to define the rule based on a validation function. The function must have already been created in the object library before it will be available in the drop-down list.
Bindings	<p>Each parameter required by the function displays. Provide an argument for each parameter. The argument can be a constant, column (from the input schema), substitution variable, data flow variable, or global variable. You can type the value into the cell, or select the drop-down arrow to display the available columns and variables.</p> <p>Clear the Score check box to not include the binding in the Fail output. See the following example for more information about Score.</p>
Column Validation	Select to base the rule on a single input column.
Column	Click the drop-down arrow to select a column on which to process the rule.
Condition	<p>Select a condition (usually in conjunction with an expression) to define the column-based rule. Available condition operators include:</p> <p><, >, <=, >=, <>, =</p> <p>IS NULL, IS NOT NULL</p> <p>LIKE</p> <p>IN SET</p> <p>BETWEEN: Use to specify a range of values.</p> <p>Match Pattern: Lets you enter a pattern based on the Data Services match_pattern function.</p> <p>Exists in Table: Select to specify that a column's value must exist in another table's column. Click the drop-down arrow to open the Input Parameter window and browse to the desired input. This option uses the LOOKUP_EXT function. Define the NOT NULL constraint for the column in the LOOKUP table to ensure the Exists in table condition executes properly.</p> <p>Custom Condition: Select to create more complex expressions by linking to the smart editor (ellipses button) or function wizard (Function button). An edit box opens for you to enter your expression.</p>

❖ Example

Understanding Score

Say you are using a validation function to define a validation rule that checks for valid ZIP codes and has two input parameters: \$Country and \$ZIP. The function might be written as follows:

```
if ($Country = 'US')
  if (match_pattern ($ZIP, '99999')=1)
    return 1; #For the country US, the ZIP code must match the 5-digit pattern
  else
    return 0; #Fails if ZIP code does not match pattern for US
return 1; #If country is not US, always true
```

Using this function in a Validation transform binds the rule to the columns Country and ZIP. However in the Rule Editor when defining the parameter, you would select the [Score](#) check box for the ZIP column, but not for the Country column because the goal is to validate the ZIP format. The \$Country parameter is being used as a filter (to sort out U.S. ZIP codes), not for validation.

Sample input:

Row_ID	Country	ZIP
1	US	12345
2	US	123
3	Belgium	123

Fail output with Score selected for ZIP:

Row_ID	Country	ZIP	DI_ERROR_ACTION	DI_ERROR_COLUMNS
2	US	123	F	Validation failed rule(s): ZIP

RuleViolation output:

Row_ID	DI_RULENAME	DI_COLUMNNAME
2	IsValidZip	ZIP

Therefore, if you later want to generate a report on why rows failed, only the ZIP column will appear as having had bad data, not the Country column.

Related Information

[match_pattern \[page 1164\]](#)

[lookup_ext \[page 1146\]](#)

[Smart editor](#)

8.6.10.4 Data outputs

The Validation transform can output up to three data outputs: Pass, Fail, and RuleViolation. Data outputs are based on the condition that you specify in the transform. You set the data outputs when you connect the output of the Validation transform with a Pass object, a Fail object, or both a Pass and Fail object in the workspace.

You can also load Pass and Fail data into multiple targets.

Option	Description
Pass	When you choose a Pass type data output, the output schema is identical to the input schema.

Option	Description						
Fail	<p>When you choose a Fail type data output, Data Services adds the following columns to the Fail output schema: DI_ERRORACTION and DI_ERRORCOLUMNS.</p> <ul style="list-style-type: none"> DI_ERRORACTION: This column indicates where Failed data was sent: <table> <tr> <th>Indicator</th><th>Description</th></tr> <tr> <td>B</td><td>Sent to both Pass and Fail outputs.</td></tr> <tr> <td>F</td><td>Sent to the Fail output.</td></tr> </table> <div> <p>i Note</p> <p>If you choose to send failed data to the Pass data output, Data Services does not track the results. Because Data Services does not add columns to Pass output types, you may substitute a value for failed data that you send to the Pass data output. The input schema is maintained in the Pass output.</p> </div> <ul style="list-style-type: none"> DI_ERRORCOLUMNS: This column displays all error messages for columns with failed rules. The names of input columns associated with each message are separated by colons. For example: <code><ValidationTransformName> failed rule(s): c1:c2</code> <div> <p>i Note</p> <p>If a row has conditions set for multiple columns and the Pass, Fail, and Both actions are specified for the row, then the precedence order is Fail, Both, Pass. For example, if one column's action is Send to Fail and the column fails, then the whole row is sent only to the Fail output. Other actions for other validation columns in the row are ignored.</p> </div>	Indicator	Description	B	Sent to both Pass and Fail outputs.	F	Sent to the Fail output.
Indicator	Description						
B	Sent to both Pass and Fail outputs.						
F	Sent to the Fail output.						
RuleViolation	<p>You use the RuleViolation output to capture each row that represents the unsuccessful execution of a validation rule. The following output schema columns contain data to help you understand which rule failed:</p> <ul style="list-style-type: none"> DI_ROWID: Multiple validation rules can fail on a single input row; however, the output schema Fail only emits a single row to report each failure. To get complete information about every failed rule, this column associates rows sent to the Fail output to those recorded in the RuleViolation output. To link the DI_ROWID to the actual data row, on the Validation Transform Options tab make sure the Create column DI_ROWID on Validation_Fail is selected, which includes DI_ROWID in the Fail output schema. DI_RULENAME and DI_COLUMNNAME: You can associate columns with multiple rules, and a rule can be associated with multiple columns, so each row identifies both the rule name and the column name so you can uniquely identify a particular rule-column pair. Only columns with Score selected in the Rule Editor will be added. 						

8.6.10.5 Nested schemas

The Validation transform can be used with nested schemas. You can associate any scalar column in a nested schema with a validation rule. You can use other nested columns in a validation condition as long as they share the same parents with the column on which the rule is defined. Data Services generates additional columns (DI_ERRORACTION and DI_ERRORCOLUMNS) for the Fail output target at the top level of the schema only. Columns must be expressed with fully qualified names.

8.6.11 XML_Map

The XML_Map transform is a data transform engine designed for hierarchical data. It provides functionality similar to a typical XQuery or XSLT engine. The XML_Map transform takes one or more source data sets and produces a single target data set. Flat data structures such as database tables or flat files are also supported as both source and target data sets. You can use the XML_Map transform to perform a variety of tasks. For example:

- You can create a hierarchical target data structure such as XML or IDoc from a hierarchical source data structure.
- You can create a hierarchical target data structure based on data from flat tables.
- You can create a flat target data set such as a database table from data in a hierarchical source data structure.

XML_Map transform works in two modes- Normal and Batch mode. In normal mode, data is handled on a row by row basis before sending it to the next transform. In batch mode, data is handled as block of rows, before sending it to the next transform. There are different transform icons to indicate each mode.

Parent topic: [Platform transforms \[page 833\]](#)

Related Information

[Case \[page 833\]](#)

[Data Mask \[page 836\]](#)

[DQM Microservices \[page 925\]](#)

[Map_Operation \[page 938\]](#)

[Merge \[page 942\]](#)

[Query \[page 943\]](#)

[Row_Generation \[page 970\]](#)

[SQL \[page 972\]](#)

[User-Defined \[page 975\]](#)

[Validation \[page 983\]](#)

8.6.11.1 Data inputs

Normal mode - One or more data sets. Each data set can be a hierarchical data source such as XML, IDoc, or a hierarchical output structure from a previous transform. It can also be row-based data such as a database table, spreadsheet, or flat file.

Batch mode - Single data set accepted. The data set can only be a hierarchical data source such as XML, IDoc, or a hierarchical output structure from a previous transform. It can also be row-based data such as a database table, spreadsheet, or flat file.

8.6.11.2 Data outputs

Normal mode - A single data set. The data set may be a hierarchical structure or row-based flat data.

Batch mode - A single data set. The data set is always a hierarchical structure or flat dataset. The Batch key columns become first-level columns of the row. The rest of the columns become second-level columns and cannot be edited.

8.6.11.3 Editor

Use the XML_Map editor to specify the Schema In, Schema Out, and Options for the XML_Map transform.

The areas can be resized in order to expand the area in which you are working. You can also expand and contract the columns to change the width of properties displayed in the input and output schema areas.

8.6.11.4 Options

Use batch mode to accumulate blocks of rows before you send those rows to the next transform. For example, using batch mode with SAP functions may improve performance.

You can group the data by *Batch size* and optionally by the *Batch key columns* in batch mode.

Options	Description
<i>Batch size</i>	This option specifies the maximum size of second-level rows for each batch. It could be a positive number, a global variable, or a substitution parameter. The default value of batch size is 10000.
<i>Batch key columns</i>	<p>This option specifies the input columns on which a given batch is constructed. Batch key columns become root-level columns of the output schema.</p> <p>When batch key column is selected, "<i>Input already sorted by batch key columns</i>" checkbox option is available. Checking it indicates that input is sorted, so the transform doesn't need to do sorting itself.</p>
<div><div>i Note</div><div>If the Batch key column is not specified, then <i>Batch size</i> defines the size of the block of rows.</div></div>	

8.6.11.5 Batch mode operation

When the XML_Map transform is run in batch mode, the number of rows in the first nested level is grouped by the *Batch size* and optionally by the *Batch key columns*.

When the *Batch key columns* option is selected, the input for the XML_Map transform is sorted based on *Batch key columns*. Sorting is done to detect batch key changes. If the *Batch key columns* option is not selected, then

there is no sort operation. Each batch is constructed purely based on the maximum *Batch size*. There are no first-level columns.

Based on the type of input, each batch is constructed. The batch is sent out when it reaches the maximum *Batch size*, and optionally if the *Batch key* changes.

8.6.11.5.1 Using Batch mode in XML_Map transform


Follow these steps to configure the XML_Map transform for batch mode.

i Note

Only one input is allowed for the XML_Map transform in batch mode.

1. In the XML_Map transform editor, select *Batch Mode*.

i Note

A special symbol  indicates that the XML_Map transform is in batch mode.

2. Construct each batch based on the *Batch size*, and optionally the *Batch key columns*.
3. Select the maximum *Batch size* value of second level rows for each batch.

i Note

Default value is 10000 and it could be a positive number, a global variable, or a substitution parameter.

4. If you opt for *Batch key columns*, then drag the selected column name from the input schema into the *Batch key columns* list box.

i Note

- The whole input schema moves one level down in the output schema, making it nested.
- Batch key columns become part of the output schema at the top-level. However, if you want to modify the output schema, then add downstream transforms.
- If the Batch key columns are not selected, then there is no sort operation. Each batch is constructed purely based on maximum batch size. There are no first-level columns.

5. Select the checkbox *input already sorted by batch key columns* if you need to indicate that input is sorted.

The batch is sent out when it reaches the maximum batch size, and optionally if the batch key changes.

8.6.11.6 Searching in an input or output schema

1. In the XML_Map editor *Find* tab, enter the search term in the *Find what* box or select from previous search terms in the drop-down list.
2. In the *Schemas* list, choose the schemas in which to search.

3. In the [Elements](#) list, choose the types of mappings in which to search.
4. In the [Where](#) list, choose the properties to search within.

i Note

You can search within one or all properties, but not within two or three specific properties at a time.

5. Select the [Match case](#) check box to constrain your search to the capitalization entered.
6. Click [Find](#).
The Designer searches the transform configuration for the words you specified within the constraints you defined.

i Note

The Designer searches for columns loaded into memory. If columns are not loaded into memory, you must expand the schema to load the columns into memory before clicking [Find](#) and searching for the columns.

All matches are shown in the box below the find constraints. When you click to select a table or column name, the table or column is automatically highlighted and shown in the corresponding input or output schema area.

Initially, the Designer lists the matching columns in the order that they appear within the schemas. If you are searching both schemas, the Designer lists the first match found in the input schema first and the last match found in the output schema last. You can sort the list of matches by property. Each time you click a property heading, the Designer resorts the matches, cycling through original order, ascending order, and descending order.




Arrow icons confirm column and sort type. For example, if you sort the data by the [Description](#) property and in ascending order, an “up” arrow appears next to the [Description](#) heading. Click the heading again and a “down” arrow appears to indicate that the data is now sorted in descending alpha-numeric order. Click again and the match list returns to its original sort order.

8.6.11.7 XML_Map transform input schema

The input schema area displays all input schemas for the XML Map transform. Each input schema can contain zero or more of the following elements:

- Columns
- Nested schemas

Icons preceding columns are combinations of the following graphics:

Icon	Description
	Primary key.
	Column that is not used in output mapping.
	Column that is used in output mappings.

The *Schema In* list at the top left of the query editor indicates the schema that is currently selected. As you select schemas or columns in the input schema area, the *Schema In* list displays the corresponding schema. Conversely, you can select a schema in the *Schema In* list to move easily to a required schema.

You can right-click elements in the input schema area and select the following menu commands:

Command	Applicable elements	Effect
<i>Copy</i>	Columns, schemas	Stores a copy of the selected elements in the clipboard, leaving the elements in the input schema area.
<i>Find</i>	Anywhere in the input schema area	Locates an output element with the name or description you enter.
<i>Refresh</i>	Anywhere in the input schema area	Refreshes the display of the input schema area.
<i>Parent</i>	Columns	Selects the parent schema of the selected column.
<i>Collapse</i>	Columns, schemas	Collapses a selected schema or a selected column's parent schema (to facilitate viewing/navigation).
<i>Generate DTD</i>	Root schema only	Generates a DTD format that corresponds to the structure of the selected schema (either NRDM or relational). Generates all data types as varchar.
<i>Generate XML Schema</i>	Root schema only	Generates an XML Schema that corresponds to the structure of the selected schema (either NRDM or relational). All data types match those of the selected schema.
<i>Create File Format</i>	Schemas	Creates a file format from a relational table schema. All data types match those of the original table schema.
<i>Create HDFS File Format</i>	Schemas	Creates an HDFS file format from a relational table schema. All data types match those of the original table schema.
<i>Properties</i>	Columns, schemas	Displays the properties of the selected element.

i Note

You cannot modify the element properties.




8.6.11.8 XML_Map transform output schema

The output schema area displays the schema output from the XML_Map transform. The output schema can contain one or more of the following elements:




- Columns
- Nested schemas

Icons preceding columns are combinations of the following graphics:

Icon	Description
	Primary key.

Icon	Description
	Column that has a simple mapping. A simple mapping is either a single column or an expression with no input column (that is, an expression that does not vary with input).
	Column that has a complex mapping. A complex mapping is any mapping that is not simple.
	(Red cross superimposed on any icon) Incorrect mapping.

Note

Data Services does not perform a complete validation during design, so the editor may not flag an incorrect mapping. For a complete validation, select  [Validation](#)  [Validate](#) .

The Schema Out pane shows the following:

- The current schema in the Schema Out list at the top and in the output schema area. The current schema determines:
 - The output elements that you can modify (add, map, or delete).
 - The scope of the Iteration Rule through ORDER BY tabs in the options area.
- Non-current schemas appear dim in the output schema area.

8.6.11.9 Change the current schema

There are several ways to change the current schema in a XML_Map transform:

- Select a schema from the Output list.
- Right-click a schema or column in the output schema area and select [Make Current](#).
- Double-click one of the non-current (dim) elements in the output schema area.

When you connect a target table to an XML_Map transform with an empty output schema, Data Services automatically fills the transform's output schema with the columns from the target, without mappings.

The software only fills the target schema in the output of a transform when you connect a target table to a transform with an empty output schema. If the output schema contains any column mappings, the software does not overwrite those mappings. Similarly, if you connect a transform to one target, and then disconnect that target and connect to another target, the output schema will show the columns from the first target connected.

There are several techniques to change the output schema:

- Drag and drop (or copy and paste) columns or nested schemas from the input schema area to the output schema area (this provides simple column mappings).
If you drop a column on an existing column, you can remap that column. Select [Remap Column](#) to update only the column mapping or select [Remap with Data Type](#) to update the column mapping and data type. Alternatively, you can select [Insert Above](#) or [Insert Below](#) to add the column as a new mapping or [Cancel](#) if you do not want to add the column to the output schema.
- Right-click the current schema and select [New Output Column](#) or [New Output Schema](#). You can provide simple column mappings by dragging input columns over the new output columns. For complex mappings, use the options area.
- Right-click columns in the current schema to assign and reverse primary key settings on output columns. A key icon indicates primary keys.


- Right-click the current schema and select *Unnest* to flatten output schemas. Use this command when a job has a source with a nested schema (such as an XML file), and you map columns from this source to a flat target table schema.

You can right-click elements in the output schema area and select commands. Generally, the elements must be within the current schema.

Command	Applicable elements	Effect
Cut	All	Removes the selected elements from the output schema area and stores a copy of the elements in the clipboard.
Copy	All	Stores a copy of the selected elements in the clipboard, leaving the elements in the output schema area.
Paste	All	<p>Inserts the elements stored in the clipboard at the current cursor location (this must be within the current schema). Only visible when the clipboard contains something.</p> <p>If the cursor overlaps an existing column, you are prompted to insert above, insert below, remap column, or cancel.</p>
<div> <div>i Note</div> <p>Copying an input element and pasting it in the output schema area provides a simple mapping from the input element to the output element. You can also do this by dragging the input element to the output schema area.</p> </div>		
Delete	All	Removes the selected elements from the output schema area (without making a copy).
Find...	All	Locates an output element with the name or description you enter.
Make Current	All outside the current schema	Makes the selected schema, or the schema of the selected element, the current schema.
New Output Column...	Schemas	Adds an output column to the current schema with the name and properties you enter.
New Output Schema...	Schemas	Adds a nested schema to the current schema with the name you enter.
Propagate Column From	Columns	<p>Carries a selected column schema from an upstream source or transform through intermediate objects to the output schema.</p> <p>Simple mappings are created in each object with no change to the data type or data itself.</p>
Unnest	Nested schemas	Flattens the selected schema. The selected schema disappears and a flat list of columns appears.

Command	Applicable elements	Effect
Primary Key	Columns	Toggles the primary key attribute of the column on (check mark appears next to the command) or off (no check mark appears next to the command). A key icon indicates that a column is a primary key.
Optional	All	Toggles to make a schema or column optional.
Repeatable	Nested schemas	Toggles to make a schema repeatable.
Generate DTD	Root schema only	Generates a DTD format that corresponds to the structure of the selected schema (either NRDM or relational). Generates all data types as varchar.
Generate XML Schema	Root schema only	Generates an XML Schema that corresponds to the structure of the selected schema (either NRDM or relational). All data types match those of the selected schema.
Create File Format	Nested schemas	Creates a file format from a relational table schema. All data types match those of the original table schema.
Create HDFS File Format	Nested schemas	Creates an HDFS file format from a relational table schema. All data types match those of the original table schema.
Properties	All	Displays the properties of the selected element.

8.6.11.10 Configure mappings

The mapping configuration area of the XML_Map editor contains several tabs where you enter information to specify the data you want retrieved. Tabs containing entries are flagged by a special  icon.

When you drag and drop (or copy and paste) input columns to the output schema, Data Services inserts a value in the Mapping tab. For simple mappings, this may be sufficient. For more complex mappings, complete the appropriate tabs.

Tab	Description
Mapping	Specifies how the selected output column will be derived (or mapped).
Iteration Rule	Specifies how instances for the current schema are created from instances of the source(s). An iteration rule can only be created for a repeatable target schema. Additionally, in most situations, a repeatable target schema must have an iteration rule.
WHERE	<p>Specifies conditions that determine which instances of the target schema are output.</p> <p>Enter the conditions in SQL syntax, like a WHERE clause in a SQL SELECT statement. For example:</p> <pre>TABLE1.EMPNO = TABLE2.EMPNO AND TABLE1.EMPNO > 1000 OR TABLE2.EMPNO < 9000</pre> <p>Use the Functions, Domains, and smart editor buttons for help building expressions.</p>

Tab	Description
DISTINCT	Specifies the list of distinct columns from the input or output schema (if required).
	<div> ! Restriction You cannot mix input and output columns in the DISTINCT list. </div>
GROUP BY	Specifies how the output instances are combined (if required).
ORDER BY	Specifies how the output instances are sorted (if required).
Advanced	Specifies whether to run the transform in a separate process, and defines additional options for input schemas. Options set in the Advanced tab apply to the entire XML_Map transform.
Find	Enables you to search for a specific word or term in the input and/or output schemas.

i Note

Use the WHERE through ORDER BY tabs to specify additional constraints for the current schema, similar to SQL SELECT statement clauses.

8.6.11.10.1 Mapping tab

Use the Mapping tab to specify how the selected output column is derived (or mapped). You can specify any valid expression.

Most commonly, mapping expressions contain table columns and functions.

- Enter input column names or drag columns from the input schema and drop them in the box on the Mapping tab.
- Insert functions by entering them directly, using the smart editor, or by clicking the Functions button to open the function wizard.

After you map your source to the XML_Map transform, you might determine that you need to use another transform before you send the data to the XML_Map transform. For example, you might add a validation transform to ensure that only data with a certain format is passed or you might add a Case transform to send only a subset of the data.

In general, when you change an input schema to the XML_Map transform, the Designer checks the existing top-level mappings to determine if any remapping is required.

- If the mapping contains a column with a table name that is not a current input schema name and the column is in the new input schema, the Designer automatically replaces the table name with the new input schema name. Specifically, the Designer automatically updates the input schema name for each matching column in the following mapping configuration tabs of the XML_Map editor:
 - Mapping
 - Iteration Rule
 - WHERE
 - DISTINCT
 - ORDER BY

- GROUP BY
- If the mapping contains a column that was in the obsolete table, but the column does not exist in the new input schema, you must either remove the column or remap it from the original source.

The Designer does not automatically remap the input schema for the following situations:

- When you connect a new source to the XML_Map transform before you disconnect the old source. You must click the [Schema Remapping](#) button on the Mapping tab to update the input schema name for columns in the Mapping, WHERE, GROUP BY, and ORDER BY tabs.
- When the source is a nested schema and you either change the source to a similar nested schema, or you add or delete a transform before the XML_Map transform. Click the [Schema Remapping](#) button to update the Mapping input schema name.

Schema to schema mapping

You can map a source schema to a target schema by making the target schema current and entering the source schema path in the Mapping tab. In this case, the software assumes the source and target schemas have the same structure, including the number, order and data type of columns in each level of the structure.

When you specify schema to schema mapping, you cannot also have an iteration rule, but you may specify columns in the DISTINCT, WHERE, GROUP BY, and ORDER BY tabs.

Merge To operation

When you make a target schema current in the Mapping tab, the [Merge To](#) operation becomes available. The Merge To operation allows you to copy and paste a target schema at the same level, create mappings to different sources for the copied target schema and the original schema, and then merge the result sets. In any transforms that follow the XML_Map transform, only the original target schema is displayed.

8.6.11.10.1.1 Remapping when automatic remapping was not done in the XML_Map transform

1. In the Mapping tab, click the [Schema Remapping](#) button. The [Replace Obsolete Schema window](#) opens.
2. In the [Specify obsolete schema](#) drop-down list, choose the source schema that you disconnected from the XML_Map transform.
This list displays only the top-level input schema. For an obsolete nested schema, enter the name of the top-level schema.
3. In the [Choose correct schema](#) list, choose the output schema of the transform that you added between the source and XML_Map transform.
4. Click [Remap](#).
A message displays the number of columns that were remapped; for example:

```
Schema "ODS_SALESORDER" was replaced by schema "Validation_Pass" in 11 column names.
```

8.6.11.10.2 Iteration Rule tab

Use the Iteration Rule tab to define how the output data set for the selected output schema is calculated. An iteration rule is associated only with a repeatable target node, and defines how to construct the instances of the target schema from the source data. It is a mechanism to specify the input data sets and the way the software should join them to create the target data set.

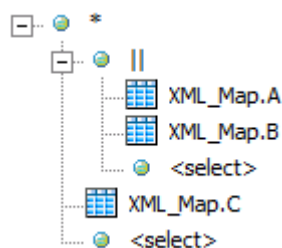
Data Services supports different kinds of joins in the iteration rule: INNER JOIN, LEFT OUTER JOIN, CROSS JOIN, and PARALLEL JOIN. PARALLEL JOIN is not a standard SQL join.

In the iteration rule tab, a hierarchical tree represents a logical combination of operations and input schemas that form a rule. Each operation in the rule is displayed as a node and may contain other operations or input schemas as children.

For example, a rule that performs a parallel operation on example tables A and B, and then combines that output set with table C by using a cross operation might logically look like this:

(A || B) * C

In the iteration rule tab, this same rule might look like this:



Constructing iteration rules

Use the iteration rule tab to create iteration rules for each repeatable schema in your output:

- To add a new element to the rule, click the `<select>` placeholder under an operation node and choose the new operation or input schema from the drop-down list.

Elements that can be added to an iteration rule include the following:

- **INNER JOIN**

Performs a SQL INNER JOIN on the sources. Create the expression to use for the join condition in the On area of the rule editor.

When you create the expression, you can use the following types of columns:

- Source columns from the sources under the current operation and the left side of the current iteration rule tree.
- Source columns from the sources that appear in the iteration rules associated with the parent schemas of the current target schema.
- Target columns from the parent schemas of the current target schema.

When using a source column, the path from the column being used to the source schema must contain no repeatable schemas.

When using a target column, it must be a scalar column and descend from the parent schema of the schema where the iteration rule is defined. In addition, the path from the parent schema to the target column must contain no repeatable schemas.

- **LEFT OUTER JOIN**

Performs a SQL LEFT OUTER JOIN on the sources. Create the expression to use for the join in the On area of the rule editor.

When you create the expression, you can use the following types of columns:

- Source columns from the sources under the current operation and the left side of the current iteration rule tree.
- Source columns from the sources that appear in the iteration rules associated with the parent schemas of the current target schema.
- Target columns from the parent schemas of the current target schema.

When using a source column, the path from the column being used to the source schema must contain no repeatable schemas.

When using a target column, it must be a scalar column and descend from the parent schema of the schema where the iteration rule is defined. In addition, the path from the parent schema to the target column must contain no repeatable schemas.

- *** - Cross operation**

Produces a Cartesian product of two or more sources. When the sources have no parent-child relationship, the behavior is the same as a standard SQL CROSS JOIN. When the sources have a parent-child relationship, the Cartesian operation provides a mechanism to iterate through all instances of the repeatable elements identified by the source schemas in the operation in the document order.

- **|| - Parallel operation**

The Parallel operation is not a standard SQL operation. It takes two or more sources and combines corresponding rows in each source to generate the output set. For example, the first rows in a pair of input tables is combined to become the first row of the output set, the second rows are combined to become the second output row, and so on.

If the sources have different numbers of rows, the output set will contain the same number of rows as the largest source. For extra rows in the output set that contain data from only one source, the additional columns that would contain data from the other sources are considered empty.

- **Available input schemas**

- To remove an element from the rule, click the element and choose **<delete>** from the drop-down list. If you remove an operation node, any child operations or schemas will also be removed from the rule.
- To change an operation type, click the operation node and choose the new operation from the drop-down list.

i Note

There is no limit to the number of sources that may be used in an iteration rule.

Automatic rule generation

The iteration rule can be generated automatically. After you have created mappings for the columns under the current target schema, click **Propose rule** in the Iteration Rule tab. The software generates the iteration rule tree. Always validate that the generated iteration rule matches your requirements. Modify the rule as needed, and add the ON condition expression when appropriate.

You can also propose rules recursively. When you click [Propose rule recursively](#), the software recursively moves through the target tree under the current target schema, finds all repeatable schemas, and generates the iteration rule for each repeatable schema based on the mappings under the schema.

i Note

Automatic rule generation is a best-guess function. For example, the software cannot know the ON condition, or whether to use INNER JOIN or LEFT OUTER JOIN. Use the automatic rule generation as a guide and always verify that the iteration rule that it creates fits your needs.

8.6.11.10.3 WHERE tab

A WHERE clause can be created for any target schema in the output structure. Use the WHERE tab to set conditions that determine which rows are output. Enter the conditions in SQL syntax, as you would a WHERE clause in a SQL SELECT statement. The WHERE tab applies to the current output schema.

You can specify any valid expression. To enter conditions, do one of the following:

- Enter expressions in the editor.
- Drag columns from the input or output schema area to the editor.
- Use the Functions button. Use the `pushdown_sql` function to have Data Services create WHERE clauses dynamically based on data rather than pre-specifying the clause.

i Note

The `pushdown_sql` function can be used if the immediate input to the XML_Map transform is the table source where you want to push the WHERE clause.

Source and target columns may be used in the WHERE expression.

! Restriction

Source columns must come from the source schemas in the current iteration rule or those that appear in the iteration rules associated with the parent schemas of the current target schema. Additionally, the path from the column being used to the source schema must contain no repeatable schemas.

Target columns must come from the current target schema or parent schemas of the current target schema. Additionally, the path from the column being used to the target schema must contain no repeatable schemas.

i Note

If your expression contains varchar comparisons, Data Services ignores trailing blanks in the data. For Oracle data, use the `rtrim` or `rpad` functions if the number of trailing blanks might differ on either side of the comparison.

8.6.11.10.4 DISTINCT tab

Use the DISTINCT tab to specify the input or output schema columns that should be used to determine whether a row is distinct. If the column specified in the DISTINCT tab contains a distinct value, the row is a new output row. The DISTINCT tab applies to the current output schema.

To add a column to the Distinct columns list, select the column in the output schema area and drag it to the box in the *DISTINCT* tab. The Designer adds the column to the bottom of the list.

To remove a column, use one of the following options:

- Right-click the column and select *Delete*.
- Select the column and click the delete icon in the top right corner of the DISTINCT tab.

To consider the entire output row as distinct, select the *Whole row is DISTINCT* option.

i Note

You cannot specify both source and target columns at the same time in the DISTINCT tab.

When source columns are used, they must descend from the source schemas in the current iteration rule. In addition, the path from the source schema to the column must contain no repeatable nodes.

When target columns are used, they must descend from the current target schema. In addition, the path from the current target schema to the column must contain no repeatable nodes.

8.6.11.10.5 GROUP BY tab

Use the GROUP BY tab to specify a list of columns for which you want to combine output. For each unique set of values in the group by list, Data Services combines or aggregates the values in the remaining columns. For example, you might want to group sales order records by order date to find the total sales ordered on a particular date. The GROUP BY tab applies to the current output schema.

To add a column to the Group By list, select the column in the input or output schema area and drag it to the box in the *GROUP BY* tab. The Designer adds the column to the bottom of the list.

The first column listed is used for primary grouping, the second column listed is used for secondary grouping, and so forth. To change the groupings, use one of the following options:

- Right-click the column and select *Move Up* or *Move Down*.
- Select the column and click the down or up arrow in the top right corner of the GROUP BY tab.

i Note

You can specify either source or target columns in the GROUP BY column list.

When source columns are used, they must descend from the source schemas in the current iteration rule. In addition, the path from the source schema to the column must contain no repeatable nodes.

When target columns are used, they must descend from the current target schema. In addition, the path from the current target schema to the column must contain no repeatable nodes.

To remove a column, use one of the following options:

- Right-click the column and select [Delete](#).
- Select the column and click the delete icon in the top right corner of the GROUP BY tab.

If you specify a group by list, then all columns in the output schema must be either in the group by list or mapped to an aggregate function, such as avg, count, max, min, or sum.

Grouping methods

While the GROUP BY operation is similar to the standard SQL GROUP BY operation, it does not always operate in exactly the same way. The XML_Map transform groups output items in different ways depending upon the columns specified and whether or not aggregation functions are used:

- **Simple grouping**
The XML_Map transform groups output items together according to the unique values of the GROUP BY list when the following conditions are met:
 - Source or target columns are specified in the GROUP BY list.
 - If source columns are specified, no aggregation functions are defined in the target schema.
 In this grouping method, the operation does not remove any items from the output data set.
- **Group aggregation**
The XML_Map transform performs grouping exactly like a standard SQL GROUP BY clause when the following conditions are met:
 - Source columns are specified in the GROUP BY list.
 - Aggregation functions are defined under the current target schema.
 - Columns in the aggregation functions descend from the sources in the current iteration rule.
 - Paths from the iterating sources to the columns do not contain any repeatable nodes.

i Note

All columns in the output schema must be either part of the group by list or mapped to an aggregate function such as avg, count, max, min, or sum.

- **Instance aggregation**
The XML_Map transform evaluates the aggregation functions for each of the items in the output data set when the following conditions are met:
 - Aggregation functions are defined under the current target schema.
 - Columns used in the aggregation functions descend from the sources in the current iteration rule.
 - Paths from the sources to the columns being used contain repeatable nodes.
 The XML_Map transform also evaluates the aggregation functions for each of the items in the output data set when the following conditions are met:
 - Aggregation functions are defined under the current target schema.
 - Columns used in the aggregation functions descend from the current target schema.
 - Paths from the current target schema to the columns being used contain repeatable nodes.

! Restriction

You cannot use both group and instance aggregation at the same time.

8.6.11.10.6 ORDER BY tab

Use the ORDER BY tab to specify the columns you want used to sort the output data set. The ORDER BY tab applies to the current output schema.

To add a column, select the column in the input or output schema area and drag it to the box on the ORDER BY tab. The Designer adds the column to the bottom of the list.

The first column listed is used for primary sorting, the second column listed is used for secondary sorting, and so forth. To change the column order, use one of the following options:

- Right-click the column and select *Move Up* or *Move Down*.
- Select the column and click the down or up arrow in the top right corner of the *ORDER BY* tab.

You can specify either source or target columns in the ORDER BY tab.

When source columns are used, they must descend from the source schemas in the current iteration rule. In addition, the path from the source schemas to the column must contain no repeatable nodes.

When target columns are used, they must descend from the current target schema. In addition, the path from the current target schema to the column must contain no repeatable nodes.

To remove a column, use one of the following options:

- Right-click the column and select *Delete*.
- Select the column and click the delete icon in the top right corner of the *ORDER BY* tab.

The default sort order is ascending. To change the order, select *Ascending* or *Descending* from the adjacent drop down box.

8.6.11.10.7 Advanced tab

Use the options in the Advanced tab to run the XML_Map transform in a separate process, or to configure options for the input schema(s). The options in the Advanced tab apply to the entire XML_Map transform.

Use the Input schemas table to configure additional options for the input schema(s) used in the transform:

Option	Description
<i>Cache</i>	<p>Indicates whether the software should read the required data from the source and load it into memory or pageable cache. Because an inner source of a join must be read for each row of an outer source, you might want to cache a source when it is used as an inner source in a join.</p> <p>Available values:</p> <ul style="list-style-type: none">• <i>Automatic</i> (default)• <i>Yes</i>• <i>No</i> <p>Cache specified in the Advanced tab of the XML_Map transform editor overrides any cache specified in a source.</p>

8.6.11.10.8 Find tab

Use the Find tab to search for a specific word or term in the input schema or the output schema.

Related Information

[Searching in an input or output schema \[page 996\]](#)

8.6.11.11 Example: Nesting data with the XML_Map transform

In this example, assume you have a database table containing a list of company employees and department information. You want to create a structure that has a list of departments, each containing a list of employees and a new column that contains the number of employees in the department.

Source	Target
<pre>Employee -departmentID -departmentName -employeeName -employedDate</pre>	<pre>Company -department (*) -departmentID -departmentName -employee (*) -employeeName -employedDate -totalEmployees</pre>

1. Create iteration rules for the `department` and `employee` schemas.
Both schemas require information from the `Employee` source table, so they must iterate on that; no JOINS or other operators are required.
2. Create a new department instance for each individual department, based on the department ID.
Include the `Company.department.departmentID` target column in the DISTINCT tab for the `department` schema. Each time a new value in that column is encountered, a new `department` instance will be created.
3. Identify and create an `employee` instance for each employee that belongs to the department.
Because the ID for the current department is already known, you can use it in an expression in the WHERE tab for the `employee` schema to include only the correct employees:

```
Employee.departmentID = Company.department.departmentID
```


When the `employee` instance iterates against the source, only rows that have the matching department ID will be selected.
4. Aggregate the number of employees in the department.
The `employee` instances have already been created, so you can use those to create a mapping expression for the `totalEmployees` column:

```
count (Company.department.employee)
```

8.6.11.12 Example: Unnesting data with the XML_Map transform

In this example, assume that you have an XML structure that contains information about purchase orders. You want to generate a flat list for the items in all of the purchase orders, ordered by the total sales for each item.

Source	Target
<pre>purchaseOrders -purchaseOrder (*) -sellerParty -sellerID -sellerName -buyerParty -buyerID -buyerName -orderLine -item (*) -name -quantity -unitPrice -currency</pre>	<pre>item -sellerID -buyerID -itemName -totalSales -currency</pre>

1. Create an iteration rule for the `item` output schema.

Because the output schema requires data from columns in multiple nested schemas, use a cross (*) operation to flatten the data.

```
*
|-purchaseOrders.purchaseOrder
|-purchaseOrders.purchaseOrder.orderLine.item
```

The input ports are always assumed to be repeatable, which means that the software expects that multiple documents of `purchaseOrders` may come in.

2. Sort the output set by the total sales for each item.

- a. Calculate the total sales amount for each item.

Use information from the source columns in an expression in the mapping for the `totalSales` column:

```
orderLine.item.quantity * orderLine.item.unitPrice
```

- b. Order the output set.

Include the `item.totalSales` target column in the ORDER BY tab for the `item` output schema.

8.6.11.13 Example: Transforming a hierarchical source to a different hierarchical target

In this example, assume that you have a hierarchical structure that contains a catalog of books. The catalog includes information associated with each book, such as name, price, quantity sold, and information about the author. You want to transform this data into a structure that is instead organized by author. You also want to calculate the total sales for each book.

Source

```

catalog
|-book
  |-Name
  |-Price
  |-Quantity
  |-Author
    |-firstName
    |-lastName
    |-Author_nt_1
      |-street
    |-city
    |-state
    |-zip

```

Target

```

authors
|-Author
  |-Name
    |-firstName
    |-lastName
  |-Address
    |-Author_nt_1
      |-street
    |-city
    |-state
    |-zip
  |-book
    |-Name
    |-Price
    |-Quantity
    |-totalSales

```

1. Create an iteration rule for the `Author` output schema.

Because the output schema requires data from columns in multiple nested schemas, use a cross (*) operation to flatten the data.

```

*
|-catalog.book
|-catalog.book.Author

```

2. Create a new `Author` instance for each individual author, based on the first and last name of the author.

Include the `authors.Author.Name.firstName` and `authors.Author.Name.lastName` target columns in the `DISTINCT` tab for the `Author` target schema. Each time a new combination of the values in those columns is encountered, a new `Author` instance will be created.

3. Sort the `Author` instances by the name of each author.

Include the `authors.Author.Name.firstName` and `authors.Author.Name.lastName` target columns in the `ORDER BY` tab for the `Author` target schema.

4. Because there may be more than one line for the street portion of the author's address, create an iteration rule for the `Author_nt_1` output schema.

```

catalog.book.Author.Author_nt_1

```

5. Map the author information from the source schema to the target schema.

Include the appropriate source column or schema in the Mapping tab for each output column or schema:

Target column or schema	Mapping expression
<code>authors.Author.Name.firstName</code>	<code>catalog.book.Author.firstName</code>
<code>authors.Author.Name.lastName</code>	<code>catalog.book.Author.lastName</code>
<code>authors.Author.Address.Author_nt_1</code>	<code>catalog.book.Author.Author_nt_1</code>
<code>authors.Author.Address.Author_nt_1.street</code>	<code>catalog.book.Author.Author_nt_1.street</code>
<code>authors.Author.Address.city</code>	<code>catalog.book.Author.city</code>
<code>authors.Author.Address.state</code>	<code>catalog.book.Author.state</code>

Target column or schema	Mapping expression
authors.Author.Address.zip	catalog.book.Author.zip

6. Create an iteration rule for the `book` output schema.

Like the `Author` output schema, the `book` output schema requires a combination of the `catalog.book` and `catalog.book.Author` source schemas. Use a cross (*) operation to flatten the data.

```
*
|-catalog.book
|-catalog.book.Author
```

7. Identify and create a `book` instance for each book that belongs to the author.

Because the first and last names of the author of the book are already known, you can use them in an expression in the WHERE tab for the `book` schema to include only the correct books:

```
catalog.book.Author.firstName = authors.Author.Name.firstName AND
catalog.book.Author.lastName = authors.Author.Name.lastName
```

8. Map book information from the source schema to the target schema.

Include the appropriate source column in the Mapping tab for each output column:

Target column	Mapping expression
authors.Author.book.Name	catalog.book.Name
authors.Author.book.Price	catalog.book.Price
authors.Author.book.Quantity	catalog.book.Quantity

9. Calculate the total sales for each book.

Use information from the source columns in an expression in the mapping for the `totalSales` column:

```
catalog.book.Price * catalog.book.Quantity
```

8.7 Text Data Processing transforms

Transforms that help you extract specific information from your text.

These transforms can parse large volumes of text, allowing you to identify and extract entities and facts, such as customers, products, locations, and financial information relevant to your organization.

[Entity Extraction transform \[page 1014\]](#)

The Entity Extraction transform extracts information from unstructured data and creates structured data that can be used by various business intelligence tools.

8.7.1 Entity Extraction transform

The Entity Extraction transform extracts information from unstructured data and creates structured data that can be used by various business intelligence tools.

Description

The Entity Extraction transform performs linguistic processing on content by using semantic and syntactic knowledge of words. You can configure the transform to identify paragraphs, sentences, and clauses and it can extract entities and facts from text. Typically, you use the Entity Extraction transform when you have text with specific information you want to extract and then use in downstream analytics and applications.

The following table provides an overview of the features in the Entity Extraction transform.

Feature	Benefit
Searches for patterns, activities, events, and relationships among entities.	The resulting data helps you to determine what type of information your data contains. Use this information for many business purposes including document and content management, data integration, business intelligence, and so on.
Understands word semantics and discovers new entities.	This understanding takes you beyond character matching for information retrieval. Use the new entities to customize extraction by creating your own entity dictionary.
Automates extraction of key information	Automation saves time by reducing the number of records to review and tag manually. It also provides insights about your data that may otherwise remain hidden in the text.

Parent topic: [Text Data Processing transforms \[page 1013\]](#)

8.7.1.1 How extraction works

The extraction process uses inherent knowledge of the semantics of words and the linguistic context in which these words occur to find entities and facts.

The extraction process creates specific patterns to extract entities and facts based on system rules. You can add entries to a dictionary and you can write custom rules to customize extraction output.

The following example shows how unstructured content is transformed into structured information for further processing and analysis.

❖ Example

Sample text and extraction information

Example of your data: "Mr. Jones is very upset with Green Insurance Corp. The offer for his totaled vehicle is too low. He states that Green offered him \$1250.00 but his car is worth anywhere from \$2500 to \$4500. Mr. Jones would like Green's comprehensive coverage to be in line with other competitors."

The Entity Extraction transform processes the text by configuring the sentiment with custom rules. It identifies the group in a logical way to identify the various parts of the entity and by identifying the facts.

The following tables show partial results with information tagged as entities, types, and subtypes.

Entity	Type	Subtype
Mr. Jones	Person	N/A
Green Insurance	Organization	Commercial
Green	Prop_Misc	N/A
\$1250.00	currency	

The following table shows partial results with information tagged as facts, types, subfacts, and subfact types.

Fact	Type	Subfact	Subfact type
Mr. Jones is very upset with Green Insurance Corp.	Sentiment	Very upset	StrongNegativeSentiment
Jones would like that Green's comprehensive coverage to be in line with other competitors.	Request	N/A	N/A

8.7.1.2 Entities, types, subtypes, and facts

The Entity Extraction transform pulls information from your data and creates entity types and subtypes. It further categorizes data into different types of facts.

Entities

In Data Services, entities are names of people, places, and things that the Entity Extraction transform extracts from your data.

❖ Example

Entities

Entity	Type
Paris	Locality
Mr. Joe Smith	Person

Subtypes

Entities can have subtypes, which are subcategories of an entity. Subtypes indicates further classification of an entity. Subtypes are hierarchical specifications of an entity type that enables the distinction between different semantic varieties of the same entity type.

❖ Example

Types and subtypes

Entity	Type	Subtype
Boeing 747	Vehicle	Air
Mercedes-Benz SL500	Vehicle	Land
SAP	Organization	Commercial

Facts

Entities by themselves only show that they are present in a document. Facts provide information about how entities are related. The term “Fact” covers multiple meanings: Complex patterns that include one or more entities, relationships between one or more entities, or predicates about an entity.

Facts indicate a pattern. The transform turns the pattern into extraction rules that extract such information as sentiments, events, or relationships. The facts provide the context about how different entities are connected in the text. The Entity Extraction transform also uses custom extraction rules that you create.

Just like entities, facts have types and subtypes.

- Fact types identify the category of a fact, such as sentiments and requests.
- Fact subtypes or subfacts are key pieces of information embedded within a fact.
- Subfact types are like categories associated with the subfact

❖ Example

Examples of Facts and fact types

Fact	Type
SAP acquired Business Objects in a friendly takeover.	Merger and acquisitions
Mr. Joe Smith is upset with is airline bookings.	Sentiment

8.7.1.3 Transform options

The Entity Extraction transform contains many options that are common to all transforms, plus specific groups of options for entity extraction.

8.7.1.3.1 Common

The [Common](#) option group includes a setting to run the transform as a separate process.

Option	Description
Run as Separate Process	Yes : Splits the transform into a separate process. No : Keeps the transform as a single, possibly multithreaded, process.

8.7.1.3.2 Languages

The [Languages](#) option group includes settings to process content in different languages, such as English, German, and French. If the input content is in a language other than the specified languages, you might see unexpected results.

Option	Description
<i>Default Language</i>	<p>Specifies the default language that the transform should assume if the Language option was <i>Auto</i> and the transform could not identify a language.</p> <p>If the value 'None' is chosen as the value for <i>Default Language</i>, and the language cannot be identified, a non-fatal error occurs and processing continues.</p> <p>If the name of some other available language is chosen as the default language, and the transform fails to identify a language on the first attempt, only entities defined for the chosen language will be extracted.</p> <p>You can use a substitution variable for this option.</p>
<i>Filter By Entity Types</i>	<p>Specifies a list of entity types (supported by the selected language) to use for filtering the extraction output.</p> <div> <p>i Note</p> <p>When the Language = 'Auto', this list is the union of all of the available entity types from the installed languages.</p> </div> <p>By default, a drop-down menu showing '...' displays. Clicking this launches the <i>Ordered Options Window - [Filter By Entity Types - Option]</i> dialog.</p> <p>Select or remove one or more entity types from the list of available entity types for that language.</p> <div> <p>i Note</p> <p>Entity type support varies among languages.</p> </div>

Option	Description
<i>Language</i>	<p>Specifies the language for processing your content. You may select another language from the list of available languages displayed alphabetically in the drop-down menu.</p> <ul style="list-style-type: none"> The default language setting is 'Auto'. You may select another language from the list, to override the automatic identification. If the transform cannot identify the language, it uses the setting of the <i>Default Language</i> option. <div style="background-color: #f0f0f0; padding: 10px; margin: 10px 0;"> <p>i Note</p> <p>If content arrives in XML, and is organized in sections, the transform identifies the language of each section.</p> </div> <p>The language name is case-insensitive.</p> <ul style="list-style-type: none"> If you select a language other than 'Auto', only entities defined for that language will be extracted for the entire document. <p>When the language is set to 'Auto', any specified dictionaries or rule files the name of which do not identify a language will always be applied. However, dictionaries and rule files that do identify a language (e.g. English) in the file name will be applied only to input identified in that language.</p> <div style="background-color: #f0f0f0; padding: 10px; margin: 10px 0;"> <p>i Note</p> <p>You will not be able to run extraction unless you have a language directory that contains the files for at least one language. By default, the language directory is installed on a client as well as a server during installation.</p> </div> <p>The default location for the language directory is: <LINK_DIR>/TextAnalysis/languages.</p> <p>You can use a substitution variable for this option.</p>

8.7.1.3.3 Processing Options

The *Processing Options* group includes configuration settings for the transform. They affect how the transform will process the content before generating the extraction output.

The *Dictionary Only* option is most useful when you want to extract entities based solely on entities defined in a dictionary. For example, you want to match exactly the product and customer names from your custom dictionary and you are not interested in any other extraction output. In such a case, getting predefined entities from the extraction process will not be of interest.

i Note

Predefined entities are entities associated with different languages and are part of the language modules. These entities are extracted by default.

The *Processing Timeout* option is most useful when you want to limit the amount of time spent on processing large content or content that takes a very long time to process.

Option	Description
<i>Advanced Parsing</i>	<p>Specifies whether advanced parsing information should be produced during extraction. Advanced parsing enriches linguistic processing including richer noun phrase structure, noun phrase coordination, and syntactic function attributes that can be leveraged in custom rules.</p> <p>This option is available only for the English language. By default, YES and NO display. If you select the YES option for non-English languages, an error message displays.</p> <p>You can use a substitution variable for this option.</p>
<i>Dictionary Only</i>	<p>Use this option to limit the extraction process to use entities defined only in the specified dictionaries. You must specify a dictionary file to use this option.</p> <div> <p>Note</p> <p>If you select this option, the extraction output does not include any predefined entities. Along with this option, if you also select the <i>Rule</i> option, the extraction output includes entities and facts defined in the rules along with entities from the specified dictionaries.</p> </div> <p>You can use a substitution variable for this option.</p>
<i>Document Properties</i>	<p>Specifies whether document properties of a binary document should be extracted, if they are present in the document. A value of YES causes the extraction, and a value of NO (the default) causes no extraction.</p> <p>Document properties are name-value pairs. The Entity Extraction transform extracts only the following document properties for binary documents:</p> <ul style="list-style-type: none"> • <APP_NAME>: The name of the software that created the document • <APP_VERSION>: The version of the software that created the document • <AUTHOR>: The name of the person who created the document • <COMPANY>: The name of the company that owns the document • <TITLE>: The title of the document • <DATE_CREATED>: The date on which the document was created <p>Document properties, if available, are extracted as entities. The <SOURCE> for the properties is called <DOC_PROPERTY> and only the following fields are defined for <DOC_PROPERTY> entities:</p> <ul style="list-style-type: none"> • <ID>: The entity ID of the document property • <SOURCE>: DOC_PROPERTY • <TYPE>: The name of the document property • <STANDARD_FORM> or <SOURCE_FORM>: The value of the document property • <CONVERTED_TEXT>: The textual content of the binary document <p>Any other output columns are not applicable to <DOC_PROPERTY> extraction rows, and have their value set to -1.</p> <p>You can use a substitution variable for this option.</p>

Option	Description
<i>Linguistic Markup</i>	<p>Specifies whether only entities, only parts of speech, or only stems, or all three are returned.</p> <ul style="list-style-type: none"> • <code><ENTITY_EXTRACTION></code> returns entities only. This is the default. • <code><POS_AND_STEM></code> returns parts of speech and stems. • <code><ENTITY_POS_AND_STEM></code> returns entities, parts of speech, and stems. <p>For more information about parts of speech tagging and stemming, see the <i>Text Data Processing Language Reference Guide</i>.</p>
<i>Processing Timeout</i>	<p>Stops processing the content after a set amount of time. 60 seconds is the default. Other values include the following:</p> <ul style="list-style-type: none"> -1 indicates no timeout should be enforced. >=1 indicates the amount of time (in seconds) after which processing should abort. <p>You can use a substitution variable for this option.</p>

8.7.1.3.4 Dictionaries

The *Dictionaries* option group includes settings to process content by specifying one or more dictionaries that should be used when performing extraction. It also enables filtering by entity types defined in each dictionary.

The *Dictionaries* option group is comprised of individual dictionaries. You can configure the transform to use multiple dictionaries. These options are found under [Dictionaries](#) > [Dictionary](#) > [Dictionary File](#).

Option	Description
<i>Dictionary</i>	<p>Use this option to add dictionaries that should be used during extraction or delete an existing dictionary. Right-click this option and select the option to duplicate an entry or to delete an entry.</p> <p>Once the entry is duplicated, change the duplicate dictionary file by selecting the dictionary to use from the directory structure.</p>
<i>Dictionary file</i>	<p>Use the <i>Browse</i> option under the drop-down menu to select a valid, compiled dictionary file to use for extraction.</p>

i Note

To include the dictionaries during extraction, they need to be accessible to the job server. If the dictionary files are located on a remote computer, include the path to those files (that can be resolved by the job server).

You can use a substitution variable for this option.

i Note

If `DICTIONARY_FILE` is defined by a substitution parameter, the transform option `DIC-
TIONARY_ENTITY_TYPES` behaves as if `DICTIONARY_FILE` is not specified.

Option	Description
Filter By Entity Types	<p>Specifies a list of entity types (defined in the selected dictionary) to use for filtering the extraction output.</p> <p>By default, a drop-down menu showing '...' displays. Clicking this launches the Ordered Options Window - [Filter By Entity Types - Option] dialog.</p> <p>Select or remove one or more entity types from the available entity list.</p>

Related Information

[Extraction dictionary \[page 1033\]](#)

8.7.1.3.5 Rules

The [Rules](#) option group includes settings to process content by specifying one or more extraction rules to use when performing extraction. It also enables filtering by rule names defined in each rule file.

The [Rule](#) option group includes individual rules. You can configure the transform to use multiple rules. These options are found under ► [Rules](#) ► [Rule](#) ► [Rule File](#) ►.

Option	Description
Filter By Rule Names	<p>Specifies a list of rule names (defined in the selected rule file) to use for filtering the extraction output.</p> <p>By default, a drop-down menu showing '...' displays. Clicking this launches the Ordered Options Window - [Filter By Rule Names - Option] dialog.</p> <p>Select or remove one or more rules from the filtering list.</p>
Rule	<p>Use this option if you want to add rules that should be used during extraction or to delete an existing rule. Right-click on this option and select the option to duplicate an entry or to delete an entry.</p> <p>Once the entry is duplicated, change the duplicate rule file by selecting the rule you want to use from the directory structure.</p>

Option	Description
Rule File	<p>Use the Browse option under the drop-down menu to select a valid, compiled rule file to use for extraction.</p> <p>To include the rules during extraction, they need to be accessible to the job server. If the rule files are located on a remote computer, include the path to those files (that can be resolved by the job server).</p> <div> <p>i Note</p> <p>A rule file typically contains multiple rules. You can use the rule filtering option to select a specific rule in a rule file.</p> </div> <p>You can use the substitution variable <code>\$\$RuleFile</code> in this option.</p> <div> <p>i Note</p> <p>If <code>RULE_FILE</code> is defined by a substitution parameter, the transform option <code>RULE_NAMES</code> behaves as if <code>RULE_FILE</code> is not specified.</p> </div>

Related Information

[Extraction rule overview \[page 1033\]](#)

8.7.1.4 Use the Entity Extraction transform

Use the Entity Extraction transform to extract information from any text, HTML, XML, or binary-formats such as PDF, and to generate structured output.

Use the generated output in several ways based on your work flow. Use it as an input to another transform or write to multiple output sources such as a database table or a flat file. The transform generates the output in UTF-16 encoding. The following list provides examples to show when you would use the Entity Extraction transform:

- To search for specific information and relationships from a large amount of text related to a broad domain. For example, a company is interested in analyzing customer feedback received in free-form text after a new product launch.
- To link structured information from unstructured text together with existing structured information to make new connections. For example, a law enforcement department is trying to make connections between various crimes and people involved using their own database and information available in various reports in text format.
- To analyze and report on product quality issues such as excessive repairs and returns for certain products by making connections between various forms of data. For example, you could have product information in structured data format such as products, parts, customers, and suppliers. Still other data contains text about product issues and customer complaints. A third data source contains maintenance records, repair logs, product escalations, and support center logs.

8.7.1.4.1 Input fields for the Entity Extraction transform

The following is a Data Services recognized input field that you can use in the input mapping for the Entity Extraction transform.

Input field name	Data type	Description												
LANGUAGE	varchar	<p>The language of the extraction.</p> <p>This can be provided at transfer run time by a dynamic input field.</p> <div>i Note<p>The following following table describes the Extraction transform behavior for various conditions of the input field <code>LANGUAGE</code>:</p><table><tr><th>Value of input field <code>LANGUAGE</code></th><th>Run-time transform behavior</th></tr><tr><td>Auto</td><td>Takes precedence over the value of the transform option <code>LANGUAGE</code></td></tr><tr><td>Valid and supported</td><td>Takes precedence over the value of the transform option <code>LANGUAGE</code></td></tr><tr><td>Valid but not supported</td><td>Error is issued and processing of input text is skipped</td></tr><tr><td>Invalid</td><td>Error is issued and processing of input text is skipped</td></tr><tr><td>Not specified</td><td>Ignored</td></tr></table></div> <p>You can use a substitution variable for this option.</p> <div>i Note<p>If <code>LANGUAGE</code> is defined by a substitution parameter, the transform option <code>LANGUAGE_ENTITY_TYPES</code> behaves as if <code>LANGUAGE</code> is set to <code>AUTO</code>.</p></div>	Value of input field <code>LANGUAGE</code>	Run-time transform behavior	Auto	Takes precedence over the value of the transform option <code>LANGUAGE</code>	Valid and supported	Takes precedence over the value of the transform option <code>LANGUAGE</code>	Valid but not supported	Error is issued and processing of input text is skipped	Invalid	Error is issued and processing of input text is skipped	Not specified	Ignored
Value of input field <code>LANGUAGE</code>	Run-time transform behavior													
Auto	Takes precedence over the value of the transform option <code>LANGUAGE</code>													
Valid and supported	Takes precedence over the value of the transform option <code>LANGUAGE</code>													
Valid but not supported	Error is issued and processing of input text is skipped													
Invalid	Error is issued and processing of input text is skipped													
Not specified	Ignored													
TEXT	Long, blob, or varchar	Mandatory field. It includes content to be processed by the transform to extract entities and/or facts. The content must be in a text format such as a text, HTML, XML, or certain binary-formats (such as PDF).												
TEXT_ID	Long, int, or varchar	<p>Optional field. Unique identifier to be used for tracing the content in case of an error.</p> <div>i Note<ul style="list-style-type: none">An unsupported data type is ignored during runtime and instead either the file name (if read from an unstructured text file format) or the string <code>TEXT</code> input field is be used as the content identifier.When a <code>varchar</code> or <code>long</code> column is mapped to the <code>TEXT_ID</code> input field and a value used to construct an error message contains more than 1K bytes, the value will be truncated to 1K.</div>												

8.7.1.4.2 Output fields for the Entity Extraction transform

The following are Data Services recognized output fields that you can use in the output mapping for the Entity Extraction transform. The fields are listed in the order they appear on the [Output](#) tab.

Output field name	Data Type	Description
CONVERTED_TEXT	long	The content text representation in UTF-16 encoding of the input text. <div>i Note When the CONVERTED_TEXT output column is selected, the first entity/fact output row for any input document will contain the UTF-16 textual representation of the input document. Any of the subsequent entity/fact output rows for the input document will not contain textual representation of the input document.</div>
ID	int	Represents a parent-child relationship between entities and/or facts. This value is unique within the scope of the processed input text. <div>i Note If you process two different input documents using the same data flow and store the output to a database, you should not use this field as a primary key.</div>
LENGTH	int	The character length of an entity or a fact in the CONVERTED_TEXT field.
OFFSET	int	The character offset of an entity or a fact in the CONVERTED_TEXT field.
PARAGRAPH_ID	int	A unique identifier of the paragraph in the CONVERTED_TEXT field containing the entity or fact.
PARENT_ID	int	Represents a parent-child relationship between entities and/or facts. If present, it provides a link to a parent ID value. If not present, this value is set to -1 to indicate there is no relationship.
SENTENCE_ID	int	A unique identifier of the sentence in the CONVERTED_TEXT field containing the entity or fact.
SOURCE	varchar (10)	The origin of an entity or fact. Meaning, how the match was determined, based on one of the following: <ul style="list-style-type: none">• SYSTEM - indicating that the entity was matched using the system files.• DICTIONARY - indicating that the entity was matched using a dictionary.• RULE - indicating that the entity or fact was matched using an extraction rule file.• LINGUISTX - indicating that the linguistics output specified in the Processing Option Linguistic Markup is displayed.
SOURCE_FORM	varchar (2000)	<ul style="list-style-type: none">• The name of an entity, fact, or subfact as mentioned in the input text.• For linguistic output, the original word in the input.

Output field name	Data Type	Description
STANDARD_FORM	varchar (2000)	<ul style="list-style-type: none"> The standard form of an entity, fact, or subfact. Generally it is the longest, most precise or official name associated with the value of the corresponding TYPE column. <div> i Note The standard form and the source form for an entity are often the same. </div> <ul style="list-style-type: none"> For linguistic output, the stem of the original word.
TYPE	varchar (255)	<ul style="list-style-type: none"> The type of an entity or fact. It may also represent subtypes or subfact types if applicable. For example, "Mr. Jones" will be identified as a PERSON entity and "car" as a COMMON_VEHICLE / LAND entity subtype. <div> i Note "/" is used as a separator to identify subtypes. </div> <ul style="list-style-type: none"> For linguistic output, the part of speech of the original word.

❖ Example

Extraction text and output fields values

The following table shows partial results of the extraction output you may see when the following sample text is processed by the Entity Extraction transform. It shows values for the following fields:

- ID
- PARENT_ID
- STANDARD_FORM
- TYPE
- SOURCE_FORM
- SOURCE

Sample input text: "Mr. Jones is very upset with Green Insurance. The offer for his totaled vehicle is too low. He states that Green offered him \$1250.00 but his car is worth anywhere from \$2500 to \$4500. Mr. Jones would like Green's comprehensive coverage to be in line with other competitors."

ID	PA- RENT_ID	STANDARD_FORM	TYPE	SOURCE_FORM	SOURCE
1	-1	Mr. Jones	PERSON	Mr. Jones	SYSTEM
2	-1	Mr. Jones is very up- set with Green In- surance.	Sentiment	Mr. Jones is very upset with Green Insurance.	RULE
3	2	Mr. Jones	Topic	Mr. Jones	RULE
4	2	very upset	StrongNegative- Sentiment	very upset	RULE
5	2	Green Insurance	Topic	Green Insurance	RULE
6	-1	Green Insurance	PROP_MISC	Green Insurance	SYSTEM

ID	PARENT_ID	STANDARD_FORM	TYPE	SOURCE_FORM	SOURCE
7	-1	The offer for his totaled vehicle is too low.	Sentiment	The offer for his totaled vehicle is too low.	RULE
8	7	totaled vehicle	Topic	totaled vehicle	RULE
9	7	too low	MinorProblem	too low	RULE
10	-1	totaled vehicle	COMMON_VEHICLE/OTHER	totaled vehicle	SYSTEM
11	-1	He states that Green offered him \$1250.00 but his car is worth anywhere from \$2500 to \$4500.	Sentiment	He states that Green offered him \$1250.00 but his car is worth anywhere from \$2500 to \$4500.	RULE
12	11	car	Topic	car	RULE
13	11	worth anywhere from \$2500 to \$4500	WeakPositiveSentiment	worth anywhere from \$2500 to \$4500	RULE
14	-1	Green	PROP_MISC	Green	SYSTEM
15	-1	\$1250.00	CURRENCY	\$1250.00	SYSTEM
16	-1	car	COMMON_VEHICLE/LAND	car	SYSTEM
17	-1	from \$2500 to \$4500	CURRENCY	from \$2500 to \$4500	SYSTEM
18	-1	Mr. Jones	PERSON	Mr. Jones	SYSTEM
19	-1	Mr. Jones would like Green's comprehensive coverage to be in line with other competitors.	Request	Mr. Jones would like Green's comprehensive coverage to be in line with other competitors.	RULE
20	19	Mr. Jones	Topic	Mr. Jones	RULE
21	19	would like	GeneralRequest	would like	RULE
22	19	Green's comprehensive coverage	Topic	Green's comprehensive coverage	RULE
23	-1	Green	PROP_MISC	Green	SYSTEM
24	-1	other competitors	COMMON_PERSON /GROUP	other competitors	SYSTEM
25	-1	other competitors	COMMON_ORGANIZATION/COMMERCIAL	other competitors	SYSTEM

In the above example, row 1 shows ID as 1 (unique identifier) with PARENT_ID as -1 (no parent relationship). The TYPE column shows entity (PERSON), facts (Sentiment), and subfacts (StrongNegativeSentiment). To

view the content of the CONVERTED_TEXT field in the Designer, use the `long_to_varchar` function to perform the conversion.

❖ Example

Extraction of linguistic information

The following table shows partial results of the extraction output you may see when the following sample text is processed by the Entity Extraction transform when you request POS_AND_STEM information:

- ID
- SOURCE_FORM
- STANDARD_FORM
- TYPE
- SOURCE

Sample input text: "The excited children boarded the airplane and flew to California, where they will visit their grandparents and many cousins."

ID	SOURCE_FORM	STANDARD_FORM	TYPE	SOURCE
1	The	the	Det-Def	LINGUISTX
2	excited	excited	Adj	LINGUISTX
3	children	child	Nn-Pl	LINGUISTX
4	boarded	board	V-Past	LINGUISTX
5	the	the	Det-Def	LINGUISTX
6	airplane	airplane	NN-Sg	LINGUISTX
7	and	and	Conj-Coord	LINGUISTX
8	flew	fly	V-Past	LINGUISTX
9	to	to	Prep	LINGUISTX
10	California	California	Prop	LINGUISTX
11	,	,	Punct-Comma	LINGUISTX
12	where	where	Conj-Sub	LINGUISTX
13	they	they	Pron	LINGUISTX
14	will	will	Aux	LINGUISTX
15	visit	visit	V-Pres	LINGUISTX
16	their	their	Det-Poss	LINGUISTX

ID	SOURCE_FORM	STANDARD_FORM	TYPE	SOURCE
17	grandparents	grandparent	Nn-Pl	LINGUISTX
18	and	and	Conj-Coord	LINGUISTX
19	many	many	Det-Pl	LINGUISTX
20	cousins	cousin	Nn-pl	LINGUISTX
21	.	.	Punct-Sent	LINGUISTX

In the above example, row 3 shows ID as 3 (unique identifier). The SOURCE_FORM shows the original word "children" as it appeared in the sentence. The STANDARD_FORM shows "child" (the stem of the source form). The TYPE identifies the word's part of speech as a plural noun. The SOURCE shows this processing was handled by the LINGUISTX analyzer.

For more information about the linguistic extraction features, see the *Text Data Processing Language Reference Guide* under "Stemming" and "Part-of-Speech Tagging".

8.7.1.4.3 Examples for using the Entity Extraction transform

This topic contains examples for using the Entity Extraction transform.

The following examples are based on a scenario in which a human resources department wants to analyze resumes that they have received in various formats. The formats include the following:

- A text file as an attachment to an e-mail
- A text resumes pasted into a field on the company Web site
- Updates to resume content that the department wants to process in real time

❖ Example

Text file e-mail attachment

The human resources department frequently receives resumes as attachments to e-mails from candidates. They store these attachments in a separate directory on a server.

To analyze and process data from these text files, they perform these steps:

1. Configure an *Unstructured text* file format that points to the directory that contains the resumes.
2. Build a data flow that contains the following objects: The unstructured text file format as the source, an Entity Extraction transform, and a target.
3. Configure the Entity Extraction transform to process and analyze the text.

❖ Example

Text resume pasted into a field on a Web site

The human resources department has an online job application form that includes a field into which applicants can paste their resumes. They then capture the field in a database table column.

To analyze and process data from the database:

1. Configure a connection to the database by creating a database datastore.
2. Build a data flow that contains the following objects: The database table as the source, an Entity Extraction transform, and a target.
3. Configure the Entity Extraction transform to process and analyze the text.

❖ Example

Updated content to be processed in real time

Suppose that the human resources department is seeking a particular qualification in an applicant. When an applicant updates their resume on the company Web-based form with the desired qualification, the HR manager wants to be immediately notified. Use a real-time job to enable this functionality.

To analyze and process the data in real time:

1. Add a real-time job to the data flow that includes begin and end markers.
2. Add the following objects to the data flow: A message source, an Entity Extraction transform, and a message target.
3. Configure the transform to process and analyze the text.

8.7.1.4.4 Using multiple transforms

Include multiple transforms in the same data flow to perform various analytics on unstructured information.

For example, to extract names and addresses embedded in some text and validate the information before running analytics on the extracted information, you could:

- Use the Entity Extraction transform to process text containing names and addresses and extract different entities.
- Pass the extraction output to the Case transform to identify which rows represent names and which rows represent addresses
- Use the Data Cleanse transform to standardize the extracted names and use the Global Address Cleanse transform to validate and correct the extracted address data.

i Note

To generate the correct data, include the `standard_form` and `type` fields in the Entity Extraction transform output schema; map the `type` field in the Case transform based on the entity type such as `PERSON`, `ADDRESS1`, etc. Next, map any `PERSON` entities from the Case transform to the Data Cleanse transform and map any `ADDRESS1` entities to the Global Address Cleanse transform.

8.7.1.4.5 Use filtering options

Filtering options enable you to limit extracted entities to specific entities defined in a dictionary, system file, rule, or a combination of all of these.

The Entity Extraction transform provides filtering options that you can use individually, or in combination so that you can control the output that is generated by the transform.

The transform provides filtering options in three categories as described in the following table.

Filter option	Option group	Description
Filter By Entity Types	Languages	Limits extraction output to include only selected entities for this language.
Filter By Entity Types	Dictionary	Limits extraction output to include only entities defined in a dictionary.
Filter By Rules Names	Rules	Limits extraction output to include only entities and facts returned by the specific rules.

The following table describes information contained in the extraction output based on the combination of the filter options.

Langues	Dictionaries	Rules	Extraction output	
Entity types	Entity types	Rule names	content	Notes
Yes	No	No	Entities, extracted using the entity types, selected in the filter.	N/A
No	Yes	No	Entities, extracted using the entity types, defined in the selected language and entity types selected from the dictionaries filter.	If multiple dictionaries are specified that contain the same entity type but this entity type is selected as a filter for only one of them, entities of this type will also be returned from the other dictionary.
Yes	Yes	No	Entities, extracted using the entity types, defined in the filters for the selected language and any specified dictionaries.	N/A

Langues Entity types	Dictionaries Entity types	Rules Rule names	Extraction output content	Notes
No	No	Yes	Entities, extracted using the entity types, defined in the selected language and any rule names selected in the filter from any specified rule files.	If multiple rule files are specified that contain the same rule name, but it is only selected as a filter for one of them, entities and facts of this type will also be returned from the other rule file.
No	Yes	Yes	Entities, extracted using entity types, defined in the selected language, entity types selected from the dictionaries filter, and any rule names selected in the filter from any specified rule files.	N/A
Yes	No	Yes	Entities, extracted using entity types, defined in the filters for the selected language and any rule names selected in the filter from any specified rule files.	N/A
Yes	Yes	Yes	Entities, extracted using entity types, defined in the filters for the selected language, entity types selected from the dictionaries filter, and any rule names selected in the filter from any specified rule files.	The extraction process filters the output using the union of the extracted entities or facts for the selected language, the dictionaries, and the rule files.

If you change your selection for the language, dictionaries, or rules, any filtering associated with that option will only be cleared by clicking the [Filter by...](#) option. You must select new filtering choices based on the changed selection.

i Note

If you are using multiple dictionaries (or rules) and have set filtering options for some of the selected dictionaries (or rules), the extraction process combines the dictionaries internally, and output is filtered

using the union of the entity types selected for each dictionary and rule names selected for each rule file. The output will identify the source as a dictionary (or rule) file and not the individual name of a dictionary (or rule) file.

i Note

If you select the *Dictionary Only* option under the *Processing Options* group, with a valid dictionary file, the entity types defined for the language are not included in the extraction output, but any extracted rule file entities and facts are included.

8.7.1.4.6 Extraction dictionary

An extraction dictionary is a user-defined repository of entities.

An extraction dictionary can store customized information about the entities in your data. For example, it can store entities composed of common words that have special meaning in your domain. It can store alphanumeric sequences such as specialized vocabulary or part numbers.

You can use an entity dictionary to store name variations in a structured way that can be accessed through the extraction process. A dictionary structure can help standardize references to an entity.

A dictionary whose name does not specify a language is language independent. Language independent means that you can use the same dictionary to store all your entities and that the same patterns are matched in documents of different languages. For example, you can use a language independent dictionary for the following purposes:

Purpose	Example
Name variation management	Such as recognizing "Bimmer" as a colloquial reference to the car make BMW.
Disambiguation of unknown entities	Such as extracting DF-21D, DF 21D, DF21D, and df-21d as alternative forms of the Dong Feng 21D rocket.
Control over entity recognition	Such as forcing Text Analysis to extract Dong Feng 21D as an entity of type WEAPON instead of extracting Dong Feng as a PERSON.

For more information about dictionaries and how to create them, see the *Text Data Processing Extraction Customization Guide*.

8.7.1.4.7 Extraction rule overview

An extraction rule defines custom patterns to extract entities, relationships, events, and larger extractions such as facts.

You write custom extraction rules to perform extraction that is customized to your specific needs.

For more information, see “Using Extraction Rules” in the *Text Data Processing Extraction Customization Guide*.

8.7.1.5 Differences between text data processing and data cleanse transforms

The Entity Extraction transform provides functionality similar to the Data Cleanse transform in certain cases, especially with respect to customization capabilities.

The documentation describes the differences between the two and which transform to use to meet your goals. The Text Data Processing Entity Extraction transform is for making sense of unstructured content and the Data Cleanse transform is for standardizing and cleansing structured data. The following table describes some of the main differences. In many cases, using a combination of Entity Extraction and Data Cleanse transforms will generate the data that is best suited for your business intelligence analyses and reports.

Criteria	Text Data Processing	Data Cleanse
Input type	Unstructured text that requires linguistic parsing to generate relevant information.	Structured data represented as fields in records.
Input size	More than 5KB of text.	Less than 5KB of text.
Input scope	Normally broad domain with many variations.	Specific data domain with limited variations.
Matching task	Content discovery, noise reduction, pattern matching, and relationship between different entities.	Dictionary lookup, pattern matching.
Potential usage	Identifies potentially meaningful information from unstructured content and extracts it into a format that can be stored in a repository.	Ensures quality of data for matching and storing into a repository such as Meta Data Management.
Output	Creates annotations about the source text in the form of entities, entity types, facts, and their offset, length, and so on. Input is not altered.	Creates parsed and standardized fields. Input is altered if desired.

9 Functions and Procedures

In Data Services, functions take input values and produce a return value. Procedures take input values and perform a set of operations without returning a specific value. Input values can be parameters passed into a data flow, values from a column of data, or variables defined inside a script. This section discusses functions and procedures, including detailed descriptions of built-in functions—the input parameters and required syntax, and the return values and data types.

9.1 About functions

9.1.1 Functions compared with transforms

Some functions can produce the same or similar values as transforms. However, functions and transforms operate in a different scope.

- Functions operate on single values, such as values in specific columns in a data set.
- Transforms operate on data sets, creating, updating, and deleting rows of data.

SAP Data Services does not support functions that include tables as input or output parameters, except functions imported from SAP Applications.

9.1.2 Operation of a function

The function's operation determines where you can call the function.

For example, the Lookup database function operates as an iterative function. The lookup function can cache information about the table and columns it is operating on between function calls. By contrast, conversion functions, such as `to_char`, operate as stateless functions. Conversion functions operate independently in each iteration. An aggregate function, such as `max`, requires a set of values to operate. Neither the `lookup` function (iterative) nor the `max` function (aggregate) can be called from a script or conditional where the context does not support how these functions operate.

The function type determines where a function can be used:

Type	Description
Aggregate	Generates a single value from a set of values. Aggregate functions, such as <code>max</code> , <code>min</code> , and <code>count</code> , use the data set specified by the expression in the Group By tab of a query. Can be called only from within a Query transform—not from custom functions or scripts.

Type	Description
<i>Iterative</i>	<p>Maintains state information from one invocation to another. The life of an iterative function's state information is the execution life of the query in which they are included. The lookup function is an iterative function.</p> <p>Can be called only from within a Query transform—not from functions or scripts.</p>
<i>Stateless</i>	<p>State information is not maintained from one invocation to the next. Stateless functions such as <code>to_char</code> or <code>month</code> can be used anywhere expressions are allowed.</p>

9.1.3 Arithmetic in date functions

The software performs some implicit data type conversions on date, time, datetime, and interval values.

Related Information

[Date arithmetic \[page 317\]](#)

9.1.4 Including functions in expressions

This sections discusses where and how you can include functions in expressions.

You can use functions in the following:

- Transforms (Query , Case, SQL, Map_Operation)
- Script objects
- Conditionals
- Other custom functions

Before you use a function, you need to know if the function operation makes sense in the expression you are creating.

For example:

- The `max` function cannot be used in a script or conditional where there is no collection of values on which to operate.
- Parameters can be output by a work flow but not by a data flow.

You can use two editors to add an existing function to an expression. These are:

- Smart editor
Embedded in other editor windows like the Script Editor, Conditional Editor, and Query Editor, the smart editor offers color coded syntax, a right-click menu, keyboard short cuts, and a list of available variables, data type formats, and functions that you can use to define a function.

- Function wizard

You can use the function wizard to define the parameters for an existing function. The function wizard offers the most help when defining complex functions.

Related Information

[Smart editor \[page 299\]](#)

9.1.4.1 Creating an expression that includes an existing function

1. Go to the script, query, or conditional editor in which you will include the expression.
2. Enable the smart editor and begin entering your expression.
3. When you want to include the function, click the *Functions* button.

The Designer opens the Select Function window of the function wizard.

4. Select a category in the *Function categories* box.

A list of functions in that category appears in the Function name box. The functions shown depend on the object you are using. For example, the functions available for ABAP data flows are a subset of those available for data flows.

In some cases, it does not make sense to use a function even though it is available. For example, the SQL function can be called in a mapping expression or a WHERE clause, but it would result in a SQL statement inside the SQL statement generated to execute a data flow.

5. Select a specific function in the *Function name* list.

A description of the function appears below the boxes.

6. Click *Next*.
7. Enter the values required by the function in the text boxes.

The page shown for each function is unique. Each page is designed to help you construct the current function. The example below shows the most common layout, however more complex functions may use different layouts.

Click in a box to see a description of the parameter at the bottom of the window. Use the down-arrow button to select input parameters. Use the smart editor button (...) to see a larger input box.

8. Click *Finish*.

The function and the parameters appear in the smart editor.

9.1.4.2 Editing an existing function call in an expression

You can edit function calls from a variety of editors. For example, you can edit them from the smart editor, script editor and so on. Using the function wizard to edit your complex function calls may save some time as

the different pieces of the function are parsed into separate arguments. Then, you can identify the options you need to change.

i Note

If your function contains specific spacing, line breaks or comments, you may not want to invoke the function wizard. Using the function wizard to edit an existing function call removes your formatting, spacing and comments. After you finish editing in the function wizard, the function text appears on one line with minimal spacing between the parameters.

1. Go to the expression that contains the function call that you want to change.
2. Right-click on the text of the function call and select *Edit Function*.
The function wizard opens.
3. Make changes to the function call, and then click *Finish*.
The function wizard closes and you can see the results of your function call.

Related Information

[Operation of a function \[page 1035\]](#)

[Including functions in expressions \[page 1036\]](#)

9.2 Built-in functions

SAP Data Services provides a set of built-in functions.

In the software, database and application functions, custom functions, and most built-in functions can be executed in parallel within the transforms in which they are used.

You can run each resource-intensive functions, such as `lookup_ext` and `count_distinct`, as a separate sub data flow that uses separate resources (memory and computer) from each other.

Related Information

[Catch error functions \[page 37\]](#)

9.2.1 Database and application functions

You can import the metadata for database and application functions and use them in Data Services applications. You can also import stored functions and procedures.

At run time, Data Services passes the appropriate information to the database or application from which the function was imported.

The metadata for a function includes the input and output parameters and their data types. If there are restrictions on data passed to the function, such as requiring uppercase values or limiting data to a specific range, you must enforce these restrictions in the input. You can either test the data before extraction or include logic in the data flow that calls the function.

Related Information

[About procedures \[page 1245\]](#)

9.3 Descriptions of built-in functions

This section describes each built-in function available in Data Services.

The following table lists the names and descriptions of functions, as well as the function's category in the function wizard and smart editor.

Note

For information about operators, functions, and transforms that you can use as push-down functions with Data Services, see SAP Note [2212730](#).

Function	Category	Description
abs	Math	Returns the absolute value of an input number.
add_months	Date	Adds a given number of months to a date.
ascii	String	Returns the decimal value of the first character for the given string using ASCII character set. If the character passed is not a valid ASCII character, -1 is returned.
avg	Aggregate	Calculates the average of a given set of values.
base64_decode	Miscellaneous	Returns the source data after decoding the base64-encoded input.
base64_encode	Miscellaneous	Returns the base64-encoded data in the engine locale character set.
before_image	Miscellaneous	Returns the before image value of a row. This function is available for the Map_Operation transform only and is applicable to UPDATE rows.
begin_delta_load	Miscellaneous	Marks the beginning of a delta load for a Replication Server real-time CDC job.
begin_initial_load	Miscellaneous	Marks the beginning of the initial load for a Replication Server real-time CDC job.

Function	Category	Description
cast	Conversion	Returns a value in the cast data type.
ceil	Math	Returns the smallest integer value greater than or equal to an input number.
chr	String	Get character representation of provided ASCII value.
concat_date_time	Date	Returns a datetime from separate date and time inputs.
copy_from_remote_system	Miscellaneous	Associate with a file location object that contains file transfer protocol information and local/remote server information. Transfer a file copy from a remote server to a local server before reading and processing the file in a work flow. You may use wildcard characters (* or ?) to select a group of files.
copy_to_remote_system	Miscellaneous	Associate with a file location object that contains file transfer protocol information and local/remote server information. Transfer the generated output file, which is output to the local server, to a remote server. You may use wildcard characters (* or ?) to select a group of files.
count	Aggregate	Counts the number of values in a table column.
count_distinct	Aggregate	Count the number of distinct non-null values in a table column.
current_configuration	Miscellaneous	Returns the name of the datastore configuration in use at runtime.
current_system_configuration	Miscellaneous	Returns the name of the system configuration in use at runtime. If no system configuration is defined, returns a NULL value.
dataflow_name	Miscellaneous	Returns the data flow name in which this call exists. If the call is not in a data flow, returns NULL.
datastore_field_value	Miscellaneous	Retrieves the value of a specified datastore field.
date_diff	Date	Returns the difference between two dates or times.
date_part	Date	Extracts a component of a given date.
day_in_month	Date	Determines the day in the month on which the given date falls.
day_in_week	Date	Determines the day in the week on which the given date falls.
day_in_year	Date	Determines the day in the year on which the given date falls.
db_type	Miscellaneous	Returns the database type of the datastore configuration in use at runtime.
db_version	Miscellaneous	Returns the database version of the datastore configuration in use at runtime.
db_database_name	Miscellaneous	Returns the database name of the datastore configuration in use at runtime.

Function	Category	Description
db_owner	Miscellaneous	Returns the real owner name for the datastore configuration that is in use at runtime.
decode	Miscellaneous	Returns an expression based on the first condition in the specified list that evaluates to TRUE.
decrypt_aes	Cryptographic	Decrypts the input string using the user-specified passphrase and key length using the AES algorithm.
decrypt_aes_ext	Cryptographic	Decrypts cipher text using the AES key generated using the specified passphrase and salt. The passphrase and salt must be the same as those used to encrypt the data.
double_metaphone	String	Encodes the input string using the Double Metaphone algorithm and returns a string.
encrypt_aes	Cryptographic	Encrypts the input string using the user-specified passphrase and key length using the AES algorithm.
encrypt_aes_ext	Cryptographic	Encrypts plain text and encodes it in base64 using the AES key generated by using the specified passphrase, salt, and key length. Given the same input, the encrypted output should be the same. The caller of this function must ensure that the space to hold encrypted text is at least 1.33 times larger than the original plain text.
error_timestamp	Miscellaneous (Can only be found when creating a script)	Returns the timestamp of the caught exception.
error_context	Miscellaneous (Can only be found when creating a script)	Returns the context of the caught exception. For example, " Session datapreview_job data flow debug_ Data-Flow Transform Debug"
error_message	Miscellaneous (Can only be found when creating a script)	Returns the error message of the caught exception.
error_number	Miscellaneous (Can only be found when creating a script)	Returns the error number of the caught exception.
exec	System	Sends a command to the operating system for execution.
extract_from_json	Conversion	<p>Extracts JSON data directly from a single column in a database table, and converts it into its internal nested relational data model (NRDM).</p> <p>To access this function, you must open the function wizard from within a new function call.</p>
extract_from_xml	Conversion	<p>Extracts XML directly from single column in a database table, and converts it into its internal nested relational data model (NRDM).</p> <p>To access this function, you must open the function wizard from within a new function call.</p>

Function	Category	Description
file_copy	Miscellaneous	Copies an existing file to a different location using the same file name or a different file name. Copies a group of files indicated by a wildcard to a different existing directory. Overwrites any existing target file(s) when overwrite flag is set to 1. The original file still exists in the original location after <code>file_copy</code> .
file_delete	Miscellaneous	Deletes an existing file, or deletes a group of files indicated by a wildcard.
file_exists	Miscellaneous	Checks to see if a given file or directory exists.
file_move	Miscellaneous	Moves an existing file to a different location using the same file name or a different file name. Moves a group of files indicated by a wildcard to a different existing directory. Overwrites any existing target file(s) when overwrite flag is set to 1. Original file does not exist in the original location after <code>file_move</code> .
fiscal_day	Date	Converts a given date into an integer value representing a day in a fiscal year.
floor	Math	Returns the largest integer value less than or equal to an input number.
gbq2file	Conversion	Optimizes performance when you export large-volume data from Google BigQuery to a local file via Google Cloud Storage.
gen_row_num_by_group	Miscellaneous	Returns group row number of the record.
gen_row_num	Miscellaneous	Returns an integer value beginning with 1 then incremented sequentially by 1 for each additional call. This function can be used to generate a column of row IDs.
gen_uuid	Miscellaneous	Returns a unique varchar string identifier.
get_domain_description	Miscellaneous	Returns the description of a value when given the domain name and the value.
get_env	Environment	Returns a value for the specified environmental variable.
get_error_filename	Environment	Returns the full path and file name for the error log.
get_file_attribute	Miscellaneous	Returns date created, date modified, or size (in bytes) of a physical file.
get_monitor_filename	Environment	Returns the full path and file name for the monitor log.
get_trace_filename	Environment	Returns the full path and file name for the trace log.
greatest	Miscellaneous	Returns greatest of the list of one or more expressions.
host_name	Miscellaneous	Returns the name of the computer on which the job is executing.
ifthenelse	Miscellaneous	Allows conditional logic in mapping and selection operations.

Function	Category	Description
index	String	Returns the index of a given word in a string.
init_cap	String	Changes the characters in a string to title case.
interval_to_char	Conversion	Returns a string representation of the interval.
is_group_changed	Miscellaneous	Returns 1 if the group is changed, 0 otherwise.
is_set_env	Environment	Verifies if the specified environment variable is set.
is_valid_date	Validation	Indicates if an expression can be converted into a valid date value.
is_valid_datetime	Validation	Indicates if an expression can be converted into a valid datetime value.
is_valid_decimal	Validation	Indicates if an expression can be converted into a valid decimal value.
is_valid_double	Validation	Indicates if an expression can be converted into a valid double value.
is_valid_int	Validation	Indicates if an expression can be converted into a valid integer value.
is_valid_real	Validation	Indicates if an expression can be converted into a valid real value.
is_valid_time	Validation	Indicates if an expression can be converted into a valid time value.
isempty	Miscellaneous	Indicates if a nested table contains data.
isweekend	Date	Indicates that a date corresponds to Saturday or Sunday.
job_name	Miscellaneous	Returns the name of the job in which the call to this function exists.
job_run_id	Miscellaneous	Retrieves the job run ID for the current job execution.
julian	Date	Converts a date to its integer Julian value, the number of days between the start of the Julian calendar and the date.
julian_to_date	Conversion	Converts a Julian value to a date.
key_generation	Database	Generates keys for the specified table, after determining the appropriate starting value.
last_date	Date	Returns the last date of the month for a given date.
least	Miscellaneous	Returns the least in a list of one or more expressions.
length	String	Returns the number of characters in a given string.
literal	String	Returns an input constant expression without interpolation. Allows you to assign a pattern to a variable without interpolation.
ln	Math	Returns the natural logarithm of the given numeric expression.

Function	Category	Description
load_from_gcs_to_gbq	Conversion	Loads data from Google Cloud Storage into Google BigQuery tables.
load_from_s3_to_redshift	Conversion	Uses the Redshift COPY command to copy data files from an Amazon Simple Storage Service (S3) bucket to a Redshift table.
load_to_xml	Conversion	Generates XML text from NRDM and loads it into a single database column. (Assumes the database supports XML text in its columns.)
local_to_utc	Date	Converts the input datetime of any time zone to Coordinated Universal Time (UTC).
log	Math	Returns the base-10 logarithm of the given numeric expression.
long_to_varchar	Conversion	Converts a data type from long to varchar.
lookup	Lookup	Finds a value in one table or file based on values in a second table or file.
lookup_ext	Lookup	Finds data from a database table, flat file, or memory datastore table.
lookup_seq	Lookup	Finds a value in one table based on values in a second table or file, and ensures that the sequence matches.
lower	String	Changes the characters in a string to lowercase.
lpad	String	Pads a string with characters from a specified pattern.
lpad_ext	String	Pads a string with logical characters from a specified pattern.
ltrim	String	Removes specified characters from the start of a string.
ltrim_blanks	String	Removes blank characters from the start of a string.
ltrim_blanks_ext	String	Removes blank and control characters from the start of a string.
mail_to	System	Sends the specified e-mail message.
match_pattern	String	Matches whole input strings to simple patterns supported by Data Services. This function does not match substrings.
match_regex	String	Matches whole input strings to the pattern that you specify with regular expressions (regular expressions based on the POSIX standard) and flags. This function does not match substrings.
max	Aggregate	Returns the maximum value from a list.
min	Aggregate	Returns the minimum value from a list.
mod	Math	Returns the remainder when one number is divided by another.
month	Date	Determines the month in which the given date falls.
num_to_interval	Conversion	Converts a numeric value to an interval.

Function	Category	Description
nvl	Miscellaneous	Replaces NULL values.
power	Math	Returns the value of the give expression to the specified power.
previous_row_value	Miscellaneous	Returns the column value of previous row.
print	String	Prints the given string to the trace log.
pushdown_sql	Miscellaneous	Allows you to create dynamic WHERE clauses.
quarter	Date	Determines the quarter in which the given date falls.
raise_exception	Miscellaneous	Calling this function causes an exception to be generated.
raise_exception_ext	Miscellaneous	Same as raise_exception, but takes a second parameter for an exit code.
rand	Math	Returns a random number between 0 and 1.
rand_ext	Math	Returns a random number between 0 and 1.
regex_replace	String	Matches the whole input string to the pattern that is specified with regular expressions (regular expressions based on the POSIX standard) and flags and replaces the matching part of the input string with the replacement string provided.
replace_substr	String	Returns a string where every occurrence of a given search string in the input is substituted by the given replacement string.
replace_substr_ext	String	Takes an input string, replaces specified occurrences of a specified sub-string with a specified replacement and returns the result. You can also use this function to search for hexadecimal or reference characters.
repository_name	Miscellaneous	Returns a database connection string and owner name. For example: beq-local.DBUser . This is the ID for the repository from which the job is run.
restore_repserver_cdb_backlogged_transactions	Miscellaneous	Restores backlogged transactions saved in the Replication Server real-time CDC datastore.
round	Math	Rounds a given number to the specified precision.
rpadd	String	Pads a string with characters from a given pattern.
rpadd_ext	String	Pads a string with logical characters from a given pattern.
rtrim	String	Removes given characters from the end of a string.
rtrim_blanks	String	Removes blank characters from the end of a string.
rtrim_blanks_ext	String	Removes blank and control characters from the end of a string.
sap_openhub_processchain_execute	SAP	Starts the process chain that extracts data from an SAP NetWeaver Business Warehouse (BW) and loads the extracted data into an Open Hub Destination table.

Function	Category	Description
sap_openhub_set_read_status	SAP	Sends the read status for the Open Hub table to SAP NetWeaver BW.
search_replace	String	Searches input parameters and replaces by matching criteria and values specified by search table.
set_cdc_checkpoint	Miscellaneous	Sets a check-point for a Microsoft SQL Server changed-data-capture (CDC method) job for data flows that run in a WHILE loop.
set_env	Environment	Sets an environmental variable temporarily to a specified value.
sleep	Miscellaneous	Suspends the execution of the data flow or work flow from where it is called.
soundex	String	Encodes the input string using the Soundex algorithm and returns a string. Use when you want to push down the function to the database-level.
sql	Database	Runs a SQL operation in the specified database.
sqrt	Math	Returns the square root of the given expression.
smtp_to	System	Sends the specified e-mail message using the SMTP protocol.
string_to_number	String	Returns the integer sum of all characters from the input string.
substr	String	Returns a specific portion of a string starting at a given point in the string.
sum	Aggregate	Calculates the sum of a given set of values.
sysdate	Date	Returns the current date as listed by the Job Server's operating system.
system_user_name	Miscellaneous	Returns the user name used to log into the Job Server's operating system.
sys time	Time	Returns the current time as listed by the operating system.
table_attribute	Miscellaneous	Retrieves the value of a specified table attribute.
to_char	Conversion	Converts a date or numeric type to a string.
to_date	Conversion	Converts a string to a date.
to_decimal	Conversion	Converts a varchar to a decimal.
to_decimal_ext	Conversion	Converts a varchar to a decimal, including precision as a parameter.
to_WKT_point	Conversion	Converts latitude and longitude to a geometry point in Well Known Text (WKT) format.
total_rows	Database	Returns the number of rows in a particular table in a datastore.
translate	String	Takes the input string and translates each character to its corresponding mapping to return the output string.

Function	Category	Description
trunc	Math	Truncates a given number to the specified precision.
truncate_table	Miscellaneous	Allows you to explicitly expunge data from a memory table.
upper	String	Changes the characters in a string to uppercase.
utc_to_local	Date	Converts the input in Coordinated Universal Time (UTC) to the desired timezone value.
varchar_to_long	Conversion	Converts a data type from varchar to long.
wait_for_file	Miscellaneous	Returns the existing files that match the input file pattern.
week_in_month	Date	Determines the week in the month in which the given date falls.
week_in_year	Date	Determines the week in the year in which the given date falls.
WL_GetKeyValue	String	Returns the value of a given keyword in Web log search strings.
word	String	Returns one word out of a string.
word_ext	String	Returns the word identified by its position in a delimited string.
workflow_name	Miscellaneous	Returns the work flow in which this call exists. Returns the name of the inner most work flow in cases where several work flows enclose this function call. If no work flow is found, returns job name.
year	Date	Determines the year in which the given date falls.

Related Information

[Catch error functions \[page 37\]](#)

9.3.1 abs

Returns the absolute value of a number.

Syntax

```
abs (<num>)
```

Return value

decimal, double, int, or real

The absolute value of the given number, `<num>`. The type of the return value is the same as the type of the original number.

Where

`<num>` The source number.

❖ Example

Function	Results
<code>abs (12.12345)</code>	<code>12.12345</code>
<code>abs (-12.12345)</code>	<code>12.12345</code>

9.3.2 add_months

Adds a given number of months to a date.

⇐ Syntax

```
add_months (<original_date>, <months_to_add>)
```

Return value

date

Where

<code><original_date></code>	Specify the starting year.month.date.
<code><months_to_add></code>	Number of months to add to the original date.

Details

The `<months_to_add>` can be any integer. If `<original_date>` is the last day of the month or if the resulting month has fewer days than the day component of `<original_date>`, then the result is the last day of the resulting month. Otherwise, the result has the same day component as `<original_date>`.

Function	Results
<code>add_months('1990.12.17', 1)</code>	<code>'1991.01.17'</code>
<code>add_months('2001.10.31', 4)</code>	<code>'2002.2.28'</code>

9.3.3 ascii

Returns decimal value of ASCII code of the first character in the input string.

Syntax
<code>ascii(<input_string>)</code>

Return Value

Int

Where

<code><input_string></code>	The source string.
-----------------------------------	--------------------

Details

Returns the decimal value of the ASCII code of the first character in the input string. Returns -1 if the first character is not a valid ASCII character.

❖ Example

Function	Results
<code>ascii('AaC')</code>	65

9.3.4 avg

Calculates the average of a given set of values.

≡ Syntax

```
avg(<value_list>)
```

Return value

decimal, double, int, or real

The calculated average of `<value_list>`. The average is calculated to the same precision as the input value.

Where

`<value_list>`

The source values for which to calculate an average, such as a table column.

Example

To calculate the average of values in the salary column of a table, use the avg function in a query:

- In the [Mapping](#) tab of the query editor, enter:

```
avg(SALARY)
```

- In the [Group By](#) tab in the query editor, specify the columns for which you want to group the salary, such as the department column. For each unique set of values in the group by list, such as each unique department, Data Services calculates the average salary.

9.3.5 base64_decode

Returns the source data after decoding the base64-encoded input.

Syntax

```
base64_decode(<base64-encoded input>, 'UTF-8')
```

Return Value

varchar or blob

Returns the source data after decoding the base64-encoded input. If the input is NULL or the size of the data is 0, Data Services returns NULL. Otherwise, it returns the base64-decoded data that conforms to RFC 2045.

Where

<code><base64-encoded input ></code>	The base64-encoded input data. Supports varchar and blob data types.
UTF-8	The code page of the output data. UTF-8 is required for Data Integrator version 11.7.3. This parameter is not required when the input data type is blob.

Related Information

[base64_encode \[page 1051\]](#)

9.3.6 base64_encode

Returns the base64-encoded data in the engine locale character set.

Syntax

```
base64_encode(<input data>, 'UTF-8')
```

Return Value

varchar or blob

Returns base64-encoded data. If the input data is NULL or the size is 0, Data Services returns NULL. Otherwise, it returns the base64-encoded data that conforms to RFC 2045.

Where

<code><input data></code>	The input data that needs to be encoded to base64. Supports varchar and blob data types.
UTF-8	The code page of the input data. UTF-8 is required for Data Integrator version 11.7.3. This parameter is not required when the input data type is blob.

Related Information

[base64_decode \[page 1051\]](#)

9.3.7 before_image

Retrieves the before image value of a row. This function is available for the Map_Operation transform only and is applicable to UPDATE rows.

i Note

You cannot use quoted strings in the parameter value for `before_image`. Because of this limitation, you cannot use global variable. For example, `before_image(T1.Col1)` is valid, but `before_image('T1.Col1')` is not.

≡ Syntax

```
before_image(<column_name>)
```

Return value

Inline

The return value is the same as the input column value. For example, if the input column (`column_name`) is an integer, the software returns an integer.

Where

<column_name>

The name of the table column.

❖ Example

```
before_image (Epml.SALARY)
```

Related Information

[Map_Operation \[page 938\]](#)

9.3.8 begin_delta_load

This function marks the beginning of a delta load for a real-time Replication Server CDC job.

This optional function is used to configure changed-data capture for use with databases that use Replication Server. It provides the source table's changed-data capture configuration in the Replication Server for the entire job. This function also prepares the Replication Server for making changed data available for the CDC reader to process. It also processes restore requests at the beginning of each iteration of the continuous work flow. If this function is not invoked explicitly in a script in a continuous work flow, then it is called implicitly at the beginning of each iteration of the continuous work flow. Because all Replication Server CDC data flows must be invoked inside one continuous work flow, `begin_delta_load` executes implicitly when the continuous work flow starts. But if you have other steps to perform before processing your CDC data flows, you can explicitly mark the beginning of a delta load inside the continuous work flow.

The guidelines for this function are as follows:

- If the job does not call the function `begin_initial_load`, then `begin_delta_load` must validate the Replication Server configuration (create CDC queue and replication definition and subscription).
- If `begin_initial_load` is called, then the `begin_delta_load` function does the necessary configuration to transition from initial load to delta load.
- The `begin_delta_load` function must be executed inside a continuous work flow. For each iteration of the continuous workflow, the `begin_delta_load` function does the following:
 - Begin the transaction
 - Communicates with child data flow CDC readers whether there is CDC data available to process
 - After all the readers successfully processed CDC data, commit the transaction
 - Repeat these steps in next iteration. Committing the transaction moves the pointer in the CDC queue in Replication Server to the next set of changed data.

Syntax

```
begin_delta_load()
```

9.3.9 begin_initial_load

This function marks the beginning of an initial load for a real-time Replication Server CDC job.

If the work flow calls `begin_initial_load`, it must be called before the `begin_delta_load` function. When `begin_initial_load` executes, it provides the source table's changed-data capture configuration for the Replication Server's real-time CDC data stores for the entire job. This function also prepares the Replication Server for capturing CDC transactions from the source system while the initial load is in progress.

When `begin_delta_load` executes subsequently in the CDC continuous work flow, it notifies the Replication Server to mark the endpoint of CDC transactions that were captured by the Replication Server during initial load. If you select *Enable Auto Correct* in the CDC reader, the reader generates automatically corrected records for all the transactions that occurred during the initial load.

Syntax

```
begin_initial_load()
```

9.3.10 cast

Explicitly converts an expression of one data type to another.

Syntax

```
Cast ('<expression>', '<data_type>')
```

Return Value

Returns the same value in `data_type`.

Where

<code><expression></code>	Input expression that needs to be cast to target data type.
<code><data_type></code>	Target data type. This must be a built-in data type and specified as a constant string, for example, 'decimal (28, 7)'. See the following Target Data Type table for syntax.

Details

The cast function explicitly converts the value of the first parameter into the built-in data type specified in the second parameter. The Cast Type Compatibility Matrix shows all explicit data type conversions that are valid for this function.

Cast type compatibility matrix

From / To	Date	Date time	Decimal	Double	Int	Interval	Real	Time	Time stamp	Varchar
Date	X	X							X	X
Date time	X	X						X	X	X
Decimal			X	X	X	X	X			X
Double			X	X	X	X	X			X
Int			X	X	X	X	X			X
Interval			X	X	X	X	X			X
Real			X	X	X	X	X			X
Time		X						X	X	X
Time stamp	X	X						X	X	X
Varchar	X	X	X	X	X	X	X	X	X	X

Target data type syntax

Data type	Syntax
varchar	'varchar(length)'

Target data type syntax

Data type	Syntax
decimal	'decimal(precision,scale)'
integer	'int'
real	'real'
double	'double'
timestamp	'timestamp'
datetime	'datetime'
date	'date'
time	'time'
interval	'interval'

❖ Example

Input	Output
<code>cast('20.3','decimal(3,1)')</code>	20.3

9.3.11 chr

Converts a decimal ASCII code to a character.

⌘ Syntax

```
chr (<integer_expression>)
```

Return Value

ASCII character

Where

<code><integer_expression></code>	Integer from 0 through 255. Returns NULL if the integer expression is not in this range.
---	--

Details

This function returns the character associated with the specified ASCII code decimal number. If you specify a value of less than 0 or greater than 255 for the `integer_expression` parameter, you will receive a NULL value. Use `chr` to insert control characters into character strings. For example, `chr (9)` can be used to insert `<tab>`.

❖ Example

Function	Results
<code>chr (65)</code>	'A'

9.3.12 ceil

Returns the smallest integer value greater than or equal to a number.

≡ Syntax

```
ceil (<num>)
```

Return value

decimal, double, int, or real

The indicated integer, cast as the same type as the original number, `<num>`.

Where

<code><num></code>	The source number.
--------------------------	--------------------

❖ Example

Function	Results
<code>ceil(12.12345)</code>	13.00000
<code>ceil(12)</code>	12
<code>ceil(-12.223)</code>	-12.000

9.3.13 concat_date_time

Returns a datetime from separate date and time inputs.

≡ Syntax

```
concat_date_time(<date>,<time>)
```

Return value

datetime

The datetime value obtained by combining the inputs.

❖ Example

```
concat_date_time(MS40."date",MS40."time")
```

9.3.14 copy_from_remote_system

Use this function to copy files from a remote server to your local server.

Use this function in a workflow script to transfer a file from a remote server into a local server. Then use the file as a source in a data flow. You can also set the workflow script to copy groups of files from a remote server to your local server. Indicate groups of files with a wildcard in the file name (either * or ? character).

The software uses the local and remote paths and transfer protocol information (FTP, SFTP, or SCP) from the named file location object.

Syntax

```
copy_from_remote_system("<file_location_object name>", "<remote_file_name>")
```

Return value

int

Returns 1 if function is successful. Returns 0 if function is not successful.

Where

<code><file_location_object name></code>	Name of file location object.
--	-------------------------------

<code><remote_file_name></code>	Name of the file to copy from the remote server.
---------------------------------------	--

Additional details

- Use this function on regular files only. For example, you cannot use this function to copy directory or symbolic links.
- Directory information is included in the file location object in such a way that this function does not copy any directories.
- This function does not recursively copy files under any sub-directories.

Example

User wants to copy a file named `prod.txt` from a remote server to a local server to use it in a data flow that includes a query transform. The file transfer protocol is set up in a file location object named `flo_ftp_1`. The user sets up a script object that contains the `copy_from_remote_system` function as follows:

Sample Code

```
copy_from_remote_system("flo_ftp_1", "prod.txt");
```

This function uses the FTP information in the file location object `flo_ftp_1` to copy the file named `prod.txt` that is located in a remote directory.

9.3.15 copy_to_remote_system

Uses information from the named file location object to copy file(s) from a local server to a remote server after the software has processed the file in a work flow. Also copies a group of files indicated by a wildcard in the file name (either * or ? character).

Use this function in a workflow script to transfer the generated output file, which is output to the local server, to a remote server. The software uses the local and remote paths and transfer protocol information (FTP, SFTP, or SCP) from the named file location object.

≡ Syntax

```
copy_to_remote_system("<file_location_object name>", "<local_file_name>")
```

Return value

int

Returns 1 if function is successful. Returns 0 if function is not successful.

Where

<code><file_location_object name></code>	Name of file location object.
--	-------------------------------

<code><local_file_name></code>	Name of the file to read from the local directory specified in the file location object.
--------------------------------------	--

Additional details

- Use this function on regular files only. For example, you cannot use this function to copy directory or symbolic links.
- Directory information is included in the file location object in such a way that this function does not copy any directories.
- This function does not recursively copy files under any sub-directories.

❖ Example

User wants to move a file after Data Services processing from a local server to a remote server using file transfer protocol.

The file name is `prod_out.txt`. The file transfer protocol is set up in a file location object named `flo_ftp_1`. To copy the file to the remote system, the user creates a script as follows:

Sample Code

```
copy_to_remote_system("flo_ftp_1", "prod_out.txt");
```

This function uses the FTP information in the file location object `flo_ftp_1` to copy the file named `prod_out.txt` that is located in the local directory.

9.3.16 count

Counts the number of values in a group.

Syntax

```
count(<column>)
```

Return value

int

The number of rows in the column that have a non-NULL value.

Where

<code><column></code>	The column in the input table in which to count values.
-----------------------------	---

Example

To determine the number of customers located in a specific sales region, use the count function with a filter defined in the [Where](#) tab of the query editor. The following WHERE clause selects the rows in the REGION column with the value TX:

```
REGION = "TX"
```

With the target column selected, enter the count function in the [Mapping](#) tab of the editor:

```
count(REGION)
```

9.3.17 count_distinct

Returns the number of distinct non-null values in a group.

Syntax

```
count_distinct(<expression>)
```

Return Value

Integer

Where

<expression>

Any valid expression of any type except NRDM or long data type.

Example

In a customer table, the customer's region is stored in a column named `REGION`. To count the number of distinct regions the customers come from, use the `count_distinct` function with a filter defined in the [Where](#) tab of the query editor. Enter the `count_distinct` function in the [Mapping](#) tab of the editor, as follows:

```
count_distinct(REGION)
```

Input

Name	Region	Country
Cust 1	East	US
Cust 2	East	US
Cust 3	West	US
Cust 4	East	France

Output

```
count_distinct(REGION) = 2
```


If you need to calculate the number of distinct regions per country, add the country column to the group by clause, as follows:

<code>count_distinct (REGION)</code>	Country
2	US
1	France

i Note

If you want to provide more resources to execute the `count_distinct` function, select [Run as a separate process](#). This option creates a separate sub data flow process for the `count_distinct` function when Data Services executes the data flow.

9.3.18 `current_configuration`

Returns the name of the datastore configuration that is in use at runtime. If the datastore does not support multiple configurations, for example, the datastore is a memory datastore, the name of the datastore is returned instead.

Syntax

```
current_configuration(ds_name)
```

Return Value

varchar

Where

<code><ds_name></code>	The name you enter when you create the datastore.
------------------------------	---

Example

Create a job and add a script with, for example, the following line.

```
print('Datastore Configuration used at runtime: [current_configuration()]')
```

Returns, for example, the following to the trace log:

```
Datastore configuration used at runtime: Test_DS
```

9.3.19 current_system_configuration

Returns the name of the system configuration used at runtime. If no system configuration is defined, returns a NULL value.

≡ Syntax

```
current_system_configuration()
```

Return Value

varchar

❖ Example

Create a job and add a script with, for example, the following line.

```
print('System Configuration used at runtime:
[current_system_configuration()]')
```

This line returns, for example, the following to the trace log:

```
System configuration used at runtime: Production
```

9.3.20 dataflow_name

Returns the data flow name in which this call exists. If the call is not in a data flow, returns NULL.

≡ Syntax

```
dataflow_name()
```

Return Value

varchar

❖ Example

```
print('Data Flow Name: [dataflow_name()]')
```

9.3.21 datastore_field_value

Retrieves the value of a specified datastore field.

Syntax

```
datastore_field_value(<ds_name>, field_name)
```

Return Value

varchar

Where

<code><ds_name></code>	The name you enter when you create the datastore.
<code><field_name></code>	The name of the field.

Details

The `<field_name>` should match the name seen in the language of the datastore. In the datastore editor click the Show ATL button to see the valid field names. If a specified field is not found or the datastore is invalid, NULL is returned. If the `<field_name>` is 'password' NULL is returned.

Example

Function	Results
<code>datastore_field_value('mssql', 'sql_server_database')</code>	<code>'DBUser'</code>

9.3.22 date_diff

Returns the difference between two dates or times.

Syntax

```
date_diff(<date1>,<date2>,'<fmt_str>')
```

Return Value

int

Where

<date1, date2>

<fmt_str> The string describing the format of the dates. Choose from the following values:

D	Day
H	Hours
M	Minutes
S	Seconds
MM	Months
YY	Years

Details

If date1 is smaller than date2, the date_diff function returns a positive value. If you want date_diff to return only a positive value, surround it with the abs() function.

Note

When using the sysdate function with date_diff, be aware that the value the sysdate function returns is a datetime value. Internally Data Services reads both the date and the time when it runs a sysdate function. The data that is used by the job depends on the data type of a particular column. For example, if the data type of a column in a query is date, Data Services uses only the date for calculations. The time data is ignored. If you change the data type to datetime, both a date and a time are used. If the data type is datetime and you don't want to use the time data, you can use to_char to truncate the timestamp from sysdate.

❖ Example

Function	Results
<code>date_diff(start_date,sysdate(),'D')</code>	The number of days between the date in the column start_date and today's date.
<code>date_diff(start_time,sys_time(),'M')</code>	The number of minutes between the time in the column start_time and the current time.

9.3.23 date_part

Extracts a component of a given date.

≡ Syntax

```
date_part(<in_date>,<fmt_str>)
```

Return Value

int

Where

`<in_date>` The input date.

`<fmt_str>` The string describing the format of the extracted part of the date. Choose from the following values:

YY	Year
MM	Month
DD	Day
HH	Hours
MI	Minutes
SS	Seconds

Details

This function takes in a datetime and extracts the component requested as an integer.

Note

Year is displayed as four digits, not two.

Example

Function	Results
<code>date_part('1990.12.31', 'YY')</code>	1990
<code>date_part('1991.01.17 23:44:30', 'SS')</code>	30

9.3.24 day_in_month

Determines the day in the month on which the input date falls.

Syntax

```
day_in_month(<date1>)
```

Return value

int

The number from 1 to 31 that represents the day in the month that `<date1>` occurs.

Where

<code><date1></code>	The source date.
----------------------------	------------------

This function extracts the day component from the date value.

❖ Example

Function	Results
<code>day_in_month(to_date('Jan 22, 1997','mon dd, yyyy'))</code>	22
<code>day_in_month(to_date('02/29/1996','mm/dd/yyyy'))</code>	29
<code>day_in_month(to_date('1996.12.31','yyyy.mm.dd'))</code>	31

9.3.25 day_in_week

Determines the day in the week on which the input date falls.

≡ Syntax

```
day_in_week(<date1>)
```

Return value

int

The number from 1 (Monday) to 7 (Sunday) that represents the day in the week that <date1> occurs.

Where

<date1>	The source date.
---------	------------------

This function allows you to categorize dates according to the day of the week the date falls upon. For example, all dates for which this function returns a "3" occur on Wednesday.

❖ Example

Function	Results
<code>day_in_week(to_date('Jan 22, 1997','mon dd, yyyy'))</code>	3 (Wednesday)

Function	Results
<code>day_in_week(to_date('02/29/1996','mm/dd/yyyy'))</code>	4 (Thursday)
<code>day_in_week(to_date('1996.12.31','yyyy.mm.dd'))</code>	2 (Tuesday)

9.3.26 day_in_year

Determines the day in the year on which the input date falls.

Syntax

```
day_in_year(<date1>)
```

Return value

int

The number from 1 to 366 that represents the day in the year that `<date1>` occurs.

Where

`<date1>` The source date.

❖ Example

Function	Results
<code>day_in_year(to_date('Jan 22, 1997','mon dd, yyyy'))</code>	22
<code>day_in_year(to_date('02/29/1996','mm/dd/yyyy'))</code>	60
<code>day_in_year(to_date('1996.12.31','yyyy.mm.dd'))</code>	366 (1996 was a leap year.)

9.3.27 db_type

Returns the database type of the datastore configuration in use at runtime.

This function is useful if your datastore has multiple configurations. For example, you can use this function in a SQL statement instead of using a constant. This allows the SQL statement to use the correct database type for each run no matter which datastore configuration is in use.

Syntax

```
db_type (<ds_name>)
```

Return Value

varchar

Possible db_type() return values for datastore types are as follows:

Datastore Types	Possible db_type() Return Value
Adapter	Adapter
Database	Attunity_Connector, DB2, Informix, Memory, Microsoft_SQL_Server, ODBC, Oracle, SAP, SAP_BW, SQL_Anywhere, SAP Sybase (for SAP ASE), Sybase_IQ, Teradata
JDE One World	DB2, Microsoft_SQL_Server, ODBC, or Oracle
JDE World	ODBC
Oracle Applications	Oracle
PeopleSoft	Microsoft_SQL_Server, or Oracle
SAP Applications	SAP
SAP BW Source	SAP
SAP BW Target	SAP_BW
Siebel	DB2, Microsoft_SQL_Server, or Oracle

Where

<ds_name>

The datastore name you enter when you create the data-store.

❖ Example

If you have a SQL transform that performs a function that has to be written differently for database types, you can tell the system what to do if the database type is Oracle.

In this example, the sql() function is used within a script.

```
IF (db_type('sales_ds') = 'Oracle')
BEGIN
  IF (db_version('sales_ds') = 'Oracle 9i')
    $sql_text = '...';
  ELSE
    $sql_text = '...';
END
Sql('sales_ds', '{ $sql_text }');
```

9.3.28 db_version

Returns the database version of the datastore configuration in use at runtime. This function is useful if your datastore has multiple configurations. For example, you can use this function in a SQL statement instead of using a constant. This allows the SQL statement to use the correct database version for each run no matter which datastore configuration is in use.

≡ Syntax

```
db_version(<ds_name>)
```

Return Value

varchar

Possible db_version() return values are:

Database type	Version
Oracle	Currently supported versions
Microsoft SQL Server	Currently supported versions
DB2 UDB	Currently supported versions
Informix IDS	Currently supported versions
SQL Anywhere	Currently supported versions
SAP ASE	Currently supported versions

Database type	Version
SAP Sybase IQ	Currently supported versions
Teradata	Currently supported versions
""	An empty string is returned for any other database type

Where

<ds_name>	The datastore name you enter when you create the datastore.
-----------	---

❖ Example

If you have a SQL transform that performs a function that has to be written differently for different versions of Oracle, you can tell the system which text to use for each database version. In this example, the sql() function is used within a script.

```
IF (db_type('sales_ds') = 'Oracle')
BEGIN
    IF (db_version('sales_ds') = 'Oracle 9i')
        $sql_text = '...';
    ELSE
        $sql_text = '...';
END
Sql('sales_ds', '{$sql_text}');
```

9.3.29 db_database_name

Returns the database name of the datastore configuration in use at runtime.

This function is useful if your datastore has multiple configurations and is accessing an MS SQL Server or SAP ASE database. For a datastore configuration that is using either of these database types, you enter a database name, when you create a datastore. This function returns that database name.

For example, master is a database name that exists in every Microsoft SQL Server and SAP ASE database. However, if you use different database names, you can use this function in, for example, a SQL statement instead of using a constant. This allows the SQL statement to use the correct database name for each run no matter which datastore configuration is in use.

This function returns an empty string for datastore configurations without MS SQL Server or SAP ASE as the Database Type.

≡, Syntax

```
db_database_name (<ds_name>)
```

Return Value

varchar

Where

<ds_name>

The datastore name you enter when you create the data-store.

❖ Example

If you have a SQL transform that performs a function that has to be written differently for different versions of database types, you can tell the system which text to use for each database version. In this example, the sql() function is used within a script.

```
IF (db_type('sales_ds') = 'DB2')
    $sql_text = '...';
ELSE
BEGIN
    IF (db_type('sales_ds') = 'Microsoft_SQL_Server')
        $db_name = db_database_name('sales_ds');
        $sql_text = '...';
END
Sql('sales_ds', '{$sql_text}');
```

9.3.30 db_owner

Returns the real owner name for the datastore configuration that is in use at runtime.

This function is useful if your datastore has multiple configurations because with multiple configurations, you can use alias owner names instead of database owner names. By using aliases instead of real owner names, you limit the amount of time it takes to port jobs to different environments.

For example, you can use this function in a SQL statement instead of using a constant. This allows the SQL statement to use the correct database owner for each run no matter which datastore configuration is in use.

≡ Syntax

```
db_owner(<ds_name>, <alias_name>)
```

Return Value

varchar

Where

ds_name

The datastore name that you entered when you created the datastore.

alias_name

The name of the alias that you created in the datastore, then mapped to the real owner name when you created a data-store configuration.

❖ Example

```
$real_owner = db_owner('sales_ds', 'sales_person');
```

9.3.31 decode

Returns an expression based on the first condition in the specified list of conditions and expressions that evaluates to TRUE.

≡ Syntax

```
decode(<condition_and_expression_list>,<default_expression>)
```

Return value

<expression> or <default_expression>

Returns the value associated with the first <condition> that evaluates to TRUE. The data type of the return value is the data type of the first <expression> in the <condition_and_expression_list> <.>

If the data type of any subsequent <expression> or the <default_expression> is not convertible to the data type of the first <expression>, Data Services produces an error at validation. If the data types are convertible but do not match, a warning appears at validation.

Where

<code><condition_and_expression_list></code>	<p>A comma-separated list of one or more pairs that specify a variable number of conditions. Each pair contains one <code><condition></code> and one <code><expression></code> separated by a comma. You must specify at least one <code><condition></code> and <code><expression></code> pair.</p> <p>The <code><condition></code> evaluates to TRUE or FALSE.</p> <p>The <code><expression></code> is the value that the function returns if the <code><condition></code> evaluates to TRUE.</p>
<code><default_expression></code>	<p>An expression that the function returns if none of the conditions in <code><condition_and_expression_list></code> evaluate to TRUE. You must specify a <code><default_expression></code>.</p>

Details

The `decode` function provides an easier way to write nested `ifthenelse` functions. In nested `ifthenelse` functions, you must write nested conditions and ensure that the parentheses are in the correct places, as the following example shows:

```
ifthenelse ((EMPNO = 1), '111',
  ifthenelse(EMPNO = 2), '222',
    ifthenelse(EMPNO = 3), '333',
      ifthenelse(EMPNO = 4), '444',
        'NO_ID'))))
```

In the `decode` function, you list the conditions, as the following example shows. Therefore, `decode` is less error prone than nested `ifthenelse` functions.

```
decode ((EMPNO = 1), '111',
  (EMPNO = 2), '222',
  (EMPNO = 3), '333',
  (EMPNO = 4), '444',
  'NO_ID')
```

To improve performance, Data Services pushes this function to the database server when possible. Thus, the database server, rather than Data Services, evaluates the `decode` function.

Use this function to apply multiple conditions when you map columns or select columns in a query. For more flexible control over conditions in a script, use the `IF` keyword in the scripting language.

If a condition compares a `varchar` value with trailing blanks, the `decode` function ignores the trailing blanks.

To compare a `NULL` value (`NULL` constant or variable that contains a `NULL` constant), use the `IS NULL` or `IS NOT NULL` operator. If you use the `Equal (=)` or `Not equal to (<>)` operator, the comparison against a `NULL` value always evaluates to `FALSE`.

❖ Example

Function	Results
<pre>decode((COUNTRY = 'FRANCE'), 'French', (COUNTRY = 'GERMANY'), 'German', (COUNTRY = 'ITALY'), 'Italian', (COUNTRY = 'TAIWAN'), 'China', (COUNTRY = 'USA'), 'America', (COUNTRY IS NULL), 'Unknown', 'Others')</pre>	If the value in the column COUNTRY is FRANCE, the value returned is French. If COUNTRY is NULL, the value returned is Unknown. If COUNTRY does not contain any of the values listed, the decode function returns the value Others.

9.3.32 decrypt_aes

This function decrypts the input string using the user-specified passphrase and key length using the AES algorithm.

≡ Syntax

```
decrypt_aes(<encrypted_input_string>,<passphrase>,<key_length_in_bits>)
```

Return value

Returns plain string as varchar.

In case of a failure, the function throws an exception of type execution error which results in termination of the job. You can catch the exception by using try/catch handlers.

If the encrypted input string is empty, then the return value is an empty string.

If the encrypted input string is NULL, then the return value is NULL.

Where

<code><encryptedinput_string></code>	A varchar input string to be decrypted.
<code><passphrase></code>	A varchar character string.
<code><key_length_in_bits></code>	An int value of 128, 192, or 256.

❖ Example

For security purposes, you should secure the passphrase in a database and read it using a `sql()` function into a local or global variable. Then you can pass the variable to the passphrase parameter.

```
#read the passphrase
from a secured source such as a database
$G_passphrase = sql('PASSWORD_DATASTORE', 'select PASSPHRASE from PASSWORD');
encrypt_aes(SOURCE.SSN,
$G_passphrase, 128);
```

Similar to other string functions, this function can be called from a custom function, in the column mapping of a Query transform or in a script in the work flow.

9.3.33 decrypt_aes_ext

This function decrypts the input string using the user-specified passphrase, salt, and key length using the AES algorithm. The passphrase and salt must be the same as those used to encrypt the data.

It generates an AES key of the specified key length using the specified passphrase and the key generation algorithm PKCS5_PBKDF2_SHA256. This key is used for decrypting the encrypted input string.

≡ Syntax

```
decrypt_aes_ext(<Varchar Encrypted_input_string>,<Varchar
Passphrase>,<Varchar Salt>,<Int Key_length_in_bits>)
```

Return value

Returns plain string as varchar.

In case of a failure, the function throws an exception of type execution error, which results in the termination of the job. You can catch the exception by using try/catch handlers.

If the encrypted input string is empty, then the return value is an empty string.

If the encrypted input string is NULL, then the return value is NULL.

If you fail to provide the same passphrase and key length used for encryption to this function, then the call does not fail but instead returns an incorrect output.

Where

<Encrypted_input_string>

A varchar input string to be decrypted.

<Passphrase>	A varchar character string with at least one character.
<Salt>	A varchar that must be exactly eight ASCII characters.
<Key_length_in_bits>	An int value of 128, 192, or 256.

❖ Example

For security purposes, you should secure the passphrase and salts in a database and read it using a sql() function into a local or global variable. Then you can pass the variable to the passphrase parameter.

```
#read the passphrase from a secured source such as a database
$G_passphrase = sql('PASSWORD_DATASTORE', 'select PASSPHRASE from PASSWORD');
$G_salt = sql('PASSWORD_DATASTORE', 'select SALT from PASSWORD');
decrypt_aes_ext(ENCRYPTED.SSN, $G_passphrase, $G_salt, 128);
```

Similar to other string functions, this function can be called from a custom function, in the column mapping of a Query transform, or in a script in the work flow.

9.3.34 double_metaphone

Encodes the input string using the Double Metaphone algorithm and returns a string.

≡ Syntax

```
double_metaphone(<input_str, alternate, return_if_empty>)
```

Return Value

varchar

Returns the string containing the double metaphone encoding of the input string. The length of the return string depends on the length of the input string, but it is always shorter than the input string.

Where

<input_str>	The source string to encode.
-------------	------------------------------

<code><alternate></code>	A flag to control how encoded strings are returned. When input as 0, return the primary encoding. If there is no primary encoding, then return null or the input string, depending on how <code><return_if_empty></code> is set. When the input is not 0, return the alternate encoding. If there is no alternate encoding, then return null or the input string, depending on how <code><return_if_empty></code> is set. When the parameter is null or invalid (non-numeric), it is defaulted to 0.
<code><return_if_empty></code>	<p>A flag to determine whether to return null or the input string when there is no encoding. When input as 0, return null. Otherwise, return the input string when there is no encoding. When the parameter is null or invalid (non-numeric), it is defaulted to 1.</p> <p>When input is empty, there is no primary or secondary encodings. When <code><return_if_empty>=0</code>, then return null. When <code><return_if_empty>=1</code>, then return the empty string.</p>

Details

Only use this function for input strings in English. Non-English characters are ignored.

When input is null, then return is null.

When the second or third parameter has an invalid value, default to 0 and 1, respectively.

❖ Example

Function	Result
<code>Print(double_metaphone('Hello',0,0);</code>	Prints the double metaphone of the word "Hello."
<code>double_metaphone(\$VAR, 1,1);</code>	If the string stored in \$VAR does not have encoding available, then return the original string.
<code>double_metaphone(\$VAR,'a','b');</code>	Returns the primary double metaphone encoding or the variable \$VAR when the primary encoding does not exist.

9.3.35 encrypt_aes

This function encrypts the input string using the user-specified passphrase and key length using the AES algorithm.

Syntax

```
encrypt_aes(<input_string>,<passphrase>,<key_length_in_bits>)
```

Return value

Returns encrypted string as varchar. The size of the encrypted string is usually twice as large as the size of plain text, therefore you must have enough space to hold the encrypted string.

In case of a failure, the function throws an exception of type execution error which results in termination of the job. You can catch the exception by using try/catch handlers.

If the input string is empty, then the return value is an encrypted string. The encrypted string is different for multiple calls of encrypt_aes() function with an empty input string.

If the input string is NULL, then the return value is NULL.

Where

<code><input_string></code>	A varchar input string to be encrypted.
<code><passphrase></code>	A varchar character string.
<code><key_length_in_bits></code>	An int value of 128, 192, or 256.

❖ Example

For security purposes, you should secure the passphrase in a database and read it using a sql() function into a local or global variable. Then you can pass the variable to the passphrase parameter.

```
#read the passphrase
from a secured source such as a database
$G_passphrase = sql('PASSWORD_DATASTORE', 'select PASSPHRASE from PASSWORD');
encrypt_aes(SOURCE.SSN,
$G_passphrase, 128);
```

Similar to other string functions, this function can be called from a custom function, in the column mapping of a Query transform, or in a script in the work flow.

9.3.36 encrypt_aes_ext

This function encrypts the input string using the user-specified passphrase, salt, and key length using the AES algorithm.

It generates an AES key of specified key length using the specified passphrase, salt, and the key generation algorithm PKCS5_PBKDF2_SHA256. This key is used for encrypting the input string.

Syntax

```
encrypt_aes_ext(<Varchar Input_string>,<Varchar Passphrase>,<Varchar  
salt>,<Int Key_length_in_bits>)
```

Return value

Returns encrypted string as base64 encoded string. The size of the encrypted string is 1.3 times larger than the size of plain text. Therefore you must have enough space to hold the encrypted string.

In case of a failure, the function throws an exception of type execution error, which results in the termination of the job. You can catch the exception by using try/catch handlers.

If the input string is empty, then the return value is empty.

If the input string is NULL, then the return value is NULL.

Where

<Input_string>	A varchar input string to be encrypted.
<Passphrase>	A varchar character string.
<Salt>	A varchar that must be exactly eight ASCII characters.
<Key_length_in_bits>	An int value of 128, 192, or 256.

❖ Example

For security purposes, you should secure the passphrase and salts in a database and read it using a sql() function into a local or global variable. Then you can pass the variable to the passphrase parameter.

```
#read the passphrase from a secured source such as a database
$G_passphrase = sql('PASSWORD_DATASTORE', 'select PASSPHRASE from PASSWORD');
$G_salt = sql('PASSWORD_DATASTORE', 'select SALT from PASSWORD');
encrypt_aes_ext(SOURCE.SSN, $G_passphrase, $G_salt, 128);
```

Similar to other string functions, this function can be called from a custom function, in the column mapping of a Query transform, or in a script in the work flow.

9.3.37 error_context

Returns the context of the caught exception.

Syntax

```
error_timestamp()
```

Return value

varchar 512

Example

```
"|Session datapreview_job|Dataflow debug_DataFlow|Transform Debug"
```

Related Information

[Catch error functions \[page 37\]](#)

9.3.38 error_message

Returns the error message of the caught exception.

Syntax

```
error_message()
```

Return value

varchar 512

Related Information

[Catch error functions \[page 37\]](#)

9.3.39 error_number

Returns the error number of the caught exception.

⌘ Syntax

```
error_number()
```

Return value

int

Related Information

[Catch error functions \[page 37\]](#)

9.3.40 error_timestamp

Returns the timestamp of the caught exception.

⌘ Syntax

```
error_timestamp()
```

Return value

timestamp

Related Information

[Catch error functions \[page 37\]](#)

9.3.41 exec

Sends a command to the operating system for execution. With this function, you can add a program to a Data Services job.

Syntax

```
exec(<command_file, parameter_list, flag>)
```

Return value

Varchar(1020)

Returns up to 1020 characters that depend on the value of <flag> .

Where

<command_file> A string that indicates the location and file name to execute. This string is relative to the Job Server location. It can be an absolute or relative path. The files and directories in the path must be accessible from the Job Server's computer.

The <command_file> can be a Windows batch file, a UNIX shell script, or a binary executable. To run other interpreted scripts, the <command_file> must be the name of the command interpreter (e.g., 'perl') and the script must be the first parameter in the <parameter_list> .

<parameter_list> A string that lists the values to pass as arguments to the command file. Separate parameters with spaces. When passing no parameters to an executable, enter an empty string ('').

<flag> An integer that specifies what information appears in the return value string, and how to respond upon error—how to respond if <command_file> cannot be executed or exits with a nonzero operating system return code.

Exec function flags			
Flag	If successful, returns:	On error:	Notes:
0	Standard output	Raises an exception: System function failure.	

Flag	If successful, returns:	On error:	Notes:
1	NULL string	Raises an exception: <code>System function failure</code> .	Use this flag to track error states in either of the following cases: <ul style="list-style-type: none"> The command never produces output The calling job does not need output
2	Standard output	NULL string	Use this flag if you do not intend to track the status of the command other than the presence or absence of output.
3	NULL string	NULL string	
4	Standard output	Error message string	Refer to "Details "
5	NULL string	Error message string	Refer to "Details "
8	The concatenation of the return code and the combined stdout and stderr (standard error).	Returns the concatenation of the return code and the combined stdout and stderr (standard error).	Refer to "Details "
256	NULL string	NULL string	Use this flag if you want your program to run independently of Data Services. <p>Unlike flags 0-8, if you use flag 256, Data Services will not wait until the command (executable program) completes before continuing with job processing. In this case, the command runs independently of Data Services and stdout, stderr, and return code cannot be returned.</p> <p>Raises an exception (<code>System function failure</code>) if the program cannot be launched (e.g., program file not found).</p>

Details

- The program that this function executes must not wait for any user input (e.g., a prompt for a password). For flags 0-8, Data Services waits for the program to complete, therefore if the program hangs for input, Data Services will hang also. For flag 256, Data Services will continue if the program hangs for input.
- For flags 4 and 5, the return value format for an error message string is:

```
'error-number: error-message-text'
```

where the first field is exactly 7 characters wide, and the second begins at index 10. If the program cannot be executed, the error number is 50307. If the program exits with a non-zero return code, the error number is 50306. The text is from Business Objects' `errormessage.txt`. For example:

```
' 50306: Function <exec> failed to execute program 'foo.exe'. Program
terminated with exit code 3.'
```

- For flag 8, the return value format is:

```
'return-code: stdout-and-stderr'
```

where the first field is exactly 7 characters wide and the second begins at index 10. The return code is produced by the program. Zero indicates success. Consult your program's documentation to determine the meaning of other codes.

For example:

- `' 0: 8 file(s) copied.'`
- `' 1: The system cannot find the file specified.'`
- `' 1: a.tmp -> /usr/tmp/a.tmp cp: *.lcl: The system cannot find the file specified.'`
- `' -2: manmix(): fatal application error.'`

The 7-character format enables you to easily extract the first field (the return code from the executed command) as a string of digits (which Data Services automatically converts to an integer wherever necessary), and the second field as a regular string. For example:

- In a script:

```
$foo = exec('foo.bat', ' ', 8);
```

```
$foo_rc = substr($foo, 1, 7);
```

```
$foo_txt = substr($foo, 10, 1020);
```

- In a data flow, map

```
exec('foo.bat', ' ', 8)
```

to an output column "foo" in a query. Then in a subsequent query, refer to that column's components in a mapping or WHERE clause. For example:

```
substr(query.foo, 1, 7);
```

```
substr(query.foo, 10, 1020);
```

Related Information

[exec Details: Use of remote shells \[page 1088\]](#)

9.3.41.1 exec Details: Use of remote shells

Use a remote shell to run a command elsewhere on the network:

- The `<command_file>` named in an `exec` call can be `'rsh'` on either Windows or UNIX systems to invoke the remote shell facility. This is a means of running a command on a machine elsewhere on the network. For example:
 - `exec('rsh', '<RemoteMachineName> <CommandToRunRemotely> <CmdArg1> <CmdArg2>', 0);`
 - `exec('rsh', '<RemoteBox> -l<RemoteUser> <RemoteCommand> <CmdArg>', 3);`Invoke the remote shell facility sparingly, as the remote connection setup, remote authentication, and increased message traffic reduce performance.
- For `<flag>` values 4, 5, and 8, the return code which Data Services receives is that of the `rsh` (or `remsh`) command (i.e., 0 if it successfully gets a remote connection and authorization, nonzero otherwise). There is no relation between this and the return value of the remote command (this is inherent in the remote shell mechanism on all the operating systems). To work around this, wrap the remote command in a `.bat` file (Windows) or shell script (UNIX) which will get the command's return code (`%errorlevel%` if Windows, `$?` if UNIX), and print it to `stdout` or `stderr`. For example:
 - `exec('rsh', '<RemoteMachineName> <remcmdWrapper>.bat <CmdArg1> <CmdArg2>', 8);`
 - `exec('rsh', '<RemoteBox> -l<RemoteUser> /usr/acta/<remcmdWrapper> <CmdArg>', 4);`
- The system administrator of the remote machine must set up access for the product user. The `.rhosts` and/or the `hosts.equiv` file must have an entry allowing this access.
 - If the remote machine is Windows, the Remote Shell Service must be running on it.
 - If the remote machine is UNIX, the Remote Shell daemon `rshd` must be running on it.Consult your operating system documentation for more information.

❖ Example

The examples below can be used with Windows or UNIX. If you were using the first two examples for UNIX, substitute `'sh'`, `'csh'`, `'ksh'`, `'bash'` or `'tcsh'` for `'cmd'`. Also, the first two examples call `'cmd'` rather than the program directly. You need to use `'cmd'` (or its equivalent) if either:

- The “command” is a built-in of the shell (e.g., `'DIR'` is not a program in Windows)
- Piping (a single `|` in an argument) occurs
- In either Windows or UNIX, the vertical bar symbol sends the output of one command to another command. Only use a vertical bar inside quotes. In Data Services, the double vertical bar symbol (`||`) concatenates strings. Only use a double vertical bar outside quotes.

Also, remember that the `'\'`, `'/'` symbols are interchangeable when using Windows. However, only `'/'` is accepted as a directory separator on UNIX.

```
exec('cmd', 'dir ' || $filename, 8);
```

```
exec('cmd', 'x:/bin/program1.exe | x:/bin/postprocess.bat', 4);
exec(SRC.PROGNAME, ARG_TBL.ARGS || ' lastArg', 2);
exec('c:\Data Services\bin\clone_and_rename.bat', TBL.FNAME, 1);
exec('C:\Perl5\bin\perl.exe', 'C:\sandbox\stats.pl 20 50 3000', 0);
```

Related Information

[exec](#) [page 1085]

9.3.42 extract_from_json

This function allows you to extract JSON data that is stored in one field of a database table into a query's output schema with Data Services' nested relational data model (NRDM) structure. Varchar data types are supported in the input column. You can use the following methods to extract data from clob and long data types.

- Data Services converts a clob data type input to varchar if you select the *Import unsupported data types* as VARCHAR of size option when you create a database datastore connection in the Datastore Editor.
- If your source uses a long data type, use the long_to_varchar function to convert data to varchar.

Syntax

```
extract_from_json(<json_column_name>, <schema_json_name>, <enable_validation>)
```

Value	Description
json_column_name	The name of the input column that contains the JSON data. The column data type must be varchar.
< schema_json_name >	<p>The name of the JSON Schema format that describes the incoming JSON data.</p> <p>You must import the metadata for this format into Data Services. Data Services displays the format names in the <i>Formats</i> tab of the object library. The input value must be a constant since Data Services needs to know the output schema at design time.</p>
enable_validation	<p>Enable a comparison of the incoming JSON data against the format you specify for <schema_json_name>. The JSON data structure and the JSON format structure must match. When this option is enabled, the data flow throws an exception if the incoming data is not valid. Enter 1 to validate. Enter 0 to disable the validation option.</p> <p>Validation is useful during development. It provides a more precise error if a problem occurs with the incoming JSON string.</p>

Where

Once you provide parameter values, the output schema of this function will match that of the JSON format specified. You can select any of the top-level columns or the two columns Data Services generates in the NRDM for output:

- You can select any number of the top-level columns from the JSON schema. The return type of each column follows that defined in the JSON schema.
- `AL_ERROR_NUM` - Returns an integer which indicates if an error occurred inside the function. A 0 indicates success and any non-zero integer indicates an error.
- `AL_ERROR_MSG` - Returns an error message if `AL_ERROR_NUM` is not 0. Otherwise, returns NULL.

❖ Example

```
extract_from_json(<cust_note_column>,<note_format>,1)
```

9.3.43 extract_from_xml

This function allows you to extract XML data that is stored in one field of a database table into a query's output schema with Data Services' nested relational data model (NRDM) structure. Varchar data types are supported in the input column. You can use the following methods to extract data from clob and long data types.

- Data Services converts a clob data type input to varchar if you select the *Import unsupported data types* as VARCHAR of size option when you create a database datastore connection in the Datastore Editor.
- If your source uses a long data type, use the `long_to_varchar` function to convert data to varchar.

≡ Syntax

```
extract_from_xml(<xml_column_name>,<schema_dtd_name>,<enable_validation>)
```

Value	Description
<code>xml_column_name</code>	The name of the input column that contains the XML text. The column data type must be varchar.
<code>< schema_dtd_name ></code>	The name of the DTD or XML Schema format that describes the incoming XML text. You must import the metadata for this format into Data Services. Data Services displays the format names in the <i>Formats</i> tab of the object library. The input value must be a constant since Data Services needs to know the output schema at design time.
<code>enable_validation</code>	Enable a comparison of the incoming XML data against the format you specify for <code><schema_dtd_name></code> . The XML data structure and the XML format structure must match. When this option is enabled, the data flow throws an exception if the incoming data is not valid. Enter 1 to validate. Enter 0 to disable the validation option. Validation is useful during development. It provides a more precise error if a problem occurs with the incoming XML.

Where

Once you provide parameter values, the output schema of this function will match that of the XML Format specified. You can select any of the top-level columns or the two columns Data Services generates in the NRDM for output:

- You can select any number of the top-level columns from XML schema. The return type of each column follows that defined in the XML schema.
- `AL_ERROR_NUM` - Returns an integer which indicates if an error occurred inside the function. A 0 indicates success and any non-zero integer indicates an error.
- `AL_ERROR_MSG` - Returns an error message if `AL_ERROR_NUM` is not 0. Otherwise, returns NULL.

❖ Example

```
extract_from_xml(<cust_note_column>,<note_format>,1)
```

9.3.44 file_copy

Copies an existing file to a different location using the same file name or a different file name. Copies a group of files indicated by a wildcard (*) in the file name to a different existing directory.

Overwrites any existing target file when the overwrite flag is set to 1. The source file still exists in the original location after `file_copy`.

Use `file_copy` on regular file types only. For example, you cannot use `file_copy` for directory file types or symbolic links.

≡ Syntax

```
file_copy(<source>,<target>,overwrite_if_exist)
```

Return Value

int

Returns 1 if the file is copied to the target location. Returns 0 if the file is not copied.

Where

<code><source></code>	<p>The absolute path and name of the file to copy. Use a wildcard (*) in the file name to copy a group of files that match the wildcard criteria.</p> <p>You must have permission to access the source file and location.</p>
<code><target></code>	<p>The absolute path for the location of the copied file.</p> <ul style="list-style-type: none">• You do not have to include a file name if you want to keep the same name as the source file.• You can include a different file name to rename the moved file. <p>If you copy a group of files using a wildcard (*), enter the absolute path for the location of the copied files.</p> <p>You must have permission to access the target file and location.</p>
<code>overwrite_if_exist</code>	<p>Enter a 0 or 1.</p> <p>0 = Do not overwrite any existing file. The software does not overwrite the file if it exists in the target location.</p> <div><p>i Note</p><p>In this case, the function return value is 0, and the software issues a warning that no files were copied to the target location.</p></div> <p>1 = Overwrite any existing file. The software automatically overwrites the file if it exists in the target location.</p> <div><p>i Note</p><p>In this case, the function return value is 1, the software copies the source file to the target location, and any existing file with the same name in the target location is overwritten.</p></div>

Additional details

- The source file still exists in its original location after `file_copy`.
- You may not use the following characters in the the source and target file name: \ / : * ? " < > | except when you use the asterisk (*) in a file name to indicate a wildcard.

❖ Example

Function	Results
<code>file_copy('C:\temp\my_list.txt', 'D:\my_lists\list_a.txt', 1)</code>	<p>Copy a file, paste it into a different location with a different file name.</p> <p>The software copies the <code>my_list.txt</code> file from the source location <code>C:\temp</code> and pastes it to the target location <code>D:\my_lists\</code> using a new name <code>list_a.txt</code>. The software automatically overwrites any existing file of the same name in the target location because the overwrite flag is set to 1.</p>
<code>file_copy('C:\temp\my_*.txt', 'D:\my_lists', 1)</code>	<p>Copy a group of files from one location and paste them into a different location.</p> <p>The software copies all files that match the wildcard file name <code>my_*.txt</code> from the source location <code>C:\temp</code> to the target location <code>D:\my_lists</code>. The software automatically overwrites any existing files of the same name in the target location because the overwrite flag is set to 1.</p>

9.3.45 file_delete

Deletes an existing file, or deletes a group of files indicated by a wildcard (*).

Use `file_delete` on regular file types only. For example, you cannot use `file_delete` for directory file types or symbolic links.

≡ Syntax

```
file_delete(<DelFileName>)
```

Return Value

int

Returns 1 if the stated file is deleted. Returns 0 if the stated file is not deleted.

Where

<DelFileName>

The absolute path and file name of an existing file to delete. Use a wildcard (*) in the file name to delete a group of files that match the wildcard criteria.

You must have permission to the file and directory before you can delete the file or group of files.

Additional details

- You may not use the following characters in the the deleted file name: \ / : * ? " < > | except when you use the asterisk (*) in a file name to indicate a wildcard.

❖ Example

Function	Results
<code>file_delete('C:\users\myfile.txt')</code>	Delete a file. The software deletes the file named <code>myfile.txt</code> from <code>C:\users</code> .
<code>file_delete('C:\users\my*.txt')</code>	Delete a group of files matching a wildcard. The software deletes all files that match the wildcard file name <code>my*.txt</code> from the <code>C:\users</code> directory.

9.3.46 file_exists

Checks to see if a given file or directory exists.

≡ Syntax

```
file_exists(<file_path>)
```

Return Value

int

Returns 1 if a file or directory is present on the disk (even if 0 bytes long), 0 otherwise.

Where

<file_path>

The file name and path, relative to where the Job Server is running. It can be an absolute or relative path.

❖ Example

Examples

Invoke sleep for one second when the file temp.msg exists in the directory called "c:".

```
while (file_exists('c:/temp.msg') = 1)
begin
    sleep(1000);
end
```

Set a variable to a file name and use the function to check whether the file exists:

```
$unix_file = '/tmp/t.cpp';
if (file_exists($unix_file)) $type = 'unix';
```

Set a variable based on the value of the function:

```
$i = file_exists('c:/autoexec.bat')
```

9.3.47 file_move

Moves an existing file to a different location using the same file name or a different file name. Moves a group of files indicated by a wildcard (*) to a different existing directory.

Overwrites any existing target file when the overwrite flag is set to 1. Source file does not exist in the original location after `file_move`.

Use `file_move` on regular file types only. For example, you cannot use `file_move` for directory file types or symbolic links.

≡ Syntax

```
file_move(<source>,<target>, overwrite_if_exist)
```

Return Value

int

Returns 1 if the file is moved to the target location. Returns 0 if the file is not moved.

Where

<code><source></code>	<p>The absolute path and name of the file to move. Use a wildcard (*) in the file name to move a group of files that match the wildcard criteria.</p> <p>You must have permission to access the source file and location.</p>
<code><target></code>	<p>The absolute path for the location of the moved file (or files). You must have permission to access the target file and location.</p>
<code>overwrite_if_exist</code>	<p>Enter a 0 or 1.</p> <p>0 = Do not overwrite any existing file. The software does not overwrite the file if it exists in the target location.</p> <div><p>i Note</p><p>In this case, the function return value is 0, and the software issues a warning that no files were moved to the target location.</p></div> <p>1 = Overwrite any existing file. The software automatically overwrites the file if it exists in the target location.</p> <div><p>i Note</p><p>In this case, the function return value is 1, the software moves the source file to the target location, and any existing file with the same name in the target location is overwritten.</p></div>

Additional details

- The source file no longer exists in the original location after `file_move`.
- You may not use the following characters in the the source and target file name: `\ / : * ? " < > |`
However, you may use the asterisk character (*) in a file name to indicate a wildcard.
- You can also use the `file_move` function to rename a file.

❖ Example

Function	Results
<pre>file_move('C:\temp\my_list.txt', 'C:\users', 1)</pre>	<p>Move a file from one folder to a different folder in the same directory.</p> <p>The software moves the file named <code>my_list.txt</code> from the <code>C:\temp</code> folder to the <code>C:\users</code> folder. After <code>file_move</code>, the source file <code>my_list.txt</code> no longer exists in the source folder. The software automatically overwrites any existing file named <code>my_list.txt</code> in the target location because the <code>overwrite</code> flag is set to 1.</p>
<pre>file_move('C:\temp\my_list.txt', 'C:\temp\my_list2.txt', 0)</pre>	<p>Rename a file.</p> <p>The software renames the file <code>my_list.txt</code> to <code>my_list2.txt</code>. After the function successfully executes, the source file <code>my_list.txt</code> no longer exists.</p> <p>In this example, the <code>overwrite_if_exist</code> flag is set to 0, which means:</p> <ul style="list-style-type: none">• If the file doesn't already exist in the target location, the function return value is 1, and the file is moved to the target location.• If the file already exists in the target location, the function return value is 0, and the software issues a warning that the file was not moved to the target location because the file already exists.
<pre>file_move('C:\temp\my*.txt', 'C:\users', 1)</pre>	<p>Move a group of files using a wildcard.</p> <p>The software moves all files that match the wildcard file name <code>my*.txt</code> from the source <code>C:\temp</code> to the target <code>C:\users</code>. After <code>file_move</code>, the files that match the wildcard file name no longer exist in the source folder. The software automatically overwrites any files that already exist in the target folder because the <code>overwrite</code> flag is set to 1.</p>

9.3.48 fiscal_day

Converts a given date into an integer value representing a day in a fiscal year.

≡ Syntax

```
fiscal_day('<start_year_date>', <in_date>)
```

Return Value

int

Where

<code><start_year_date></code>	The first month and day of a fiscal year. Use this format: 'mm.dd'.
<code><in_date></code>	The date you want to convert. Use any valid datetime.

Example

Function	Results
<code>fiscal_day('03.01', '1999.04.20')</code>	51

9.3.49 floor

Returns the largest integer value equal to or less than a number.

Syntax

```
floor(<num>)
```

Return value

decimal, double, int, or real

The indicated integer, cast as the same type as the original number, `<num>`.

Where

<code><num></code>	The source number.
--------------------------	--------------------

❖ Example

Function	Results
<code>floor(12.12345)</code>	12.00000
<code>floor(12)</code>	12
<code>floor(-12.223)</code>	-13.000

9.3.50 gbq2file

A function that optimizes software performance when you export large-volume Google BigQuery results to a user-specified file on your local machine.

The software uses information in the associated Google cloud storage (GCS) file location object to identify your GCS connection information, bucket name, and compression information.

≡ Syntax

```
gbq2file('<GBQ_datastore_name>', '<any_query_in_GBQ>', '<local_file_name>', '<file_location_object>', '<field_delimiter>', '/<numeric_row_delimiter>');
```

Return value

int

Returns 1 if function is successful. Returns 0 if function is not successful.

Where

<code><GBQ_datastore_name></code>	Name of the Google BigQuery application datastore in Data Services.
<code><any_query_in_GBQ></code>	Name of the applicable query in Google BigQuery.
<code><local_file_name></code>	Local file location and name in which to store the Google data. Should be the location of your local server.

<code><file_location_object></code>	Name of the Google Cloud Storage file location object in Data Services.
<code><field_delimiter></code>	Optional. The field delimiter to use between fields in the exported data. The default is a comma.
<code><numeric_row_delimiter></code>	Numeric value for the row delimiter For example, /013

Note
Default is 10, hex 0A.

How the function works

1. The function saves your Google BigQuery results to a temporary table in Google.
2. The function uses `export job` to export data from the temporary table to GCS.

Note

If the data is larger than 1 GB, Google exports the data in multiple files.

3. The function transfers the data from your Google Cloud Storage to the local file that you specified.
4. After the transfer is complete, the function deletes the temporary table and any files from Google Cloud Storage.

For details about creating a Google BigQuery application datastore, see the *Supplement for Google BigQuery*.

Related Information

[Google Cloud Storage protocol \[page 184\]](#)

9.3.51 `gen_row_num_by_group`

Generate a column of row IDs for each ID group in the specified column, beginning with integer value 1 and then incremented sequentially by 1. When the group is changed, the value is reset to 1.

Syntax

```
gen_row_num_by_group(<expression_list>)
```

Return Value

Integer

Where

`<expression_list>`

A list of one or more comma-separated expressions.

Details

This function groups the rows based on the value of expression(s) in each row in the natural order. It returns the row ID beginning with 1, then increments it sequentially by 1 for each row in the group.

In the example below, the Contract_ID column (shown under Input) shows miscellaneous ID numbers. When the `gen_row_num_by_group` function is applied to the ID column list, the IDs in the Contract_ID column are assigned a new ID number, with the first ID in the list assigned number 1. The ID number on the next row increases by 1 and is assigned the ID number 2. The ID in the following row also increases by an increment of 1, and is assigned the number 3 (as shown in the new Version_Num column under Output in the example).

If the expression(s) corresponds to a column of a table, that column must not be a NRDM or long type column.

This function should not be used with group by clause or any aggregate function.

Example

A typical use case of this function is to assign version numbers, which can become a part of the primary key in the table, as shown below.

Input			
Record	Contract_ID	Revised_by	Revision_date
record 1	1	John	1/1/2005
record 2	1	Mary	1/15/2005
record 3	1	Tim	2/1/2005
record 4	2	Joe	2/24/05

Input			
Record	Contract_ID	Revised_by	Revision_date
record 5	2	Sue	2/30/05

A `version = gen_row_num_by_group (Contract_ID)` would give an order of `Contract_ID` to group the contracts:

Output				
Record	Contract_ID	Version_Num	Revised_by	Revision_date
record 1	1	1	John	1/1/2005
record 2	1	2	Mary	1/15/2005
record 3	1	3	Tim	2/1/2005
record 4	2	1	Joe	2/24/05
record 5	2	2	Sue	2/30/05

9.3.52 gen_row_num

Returns an integer value beginning with 1 then incremented sequentially by 1 for each additional call. This function can be used to generate a column of row IDs.

Syntax

```
gen_row_num()
```

Return Value

int

Details

Each occurrence of the function in a data flow is a unique instance, resulting in a unique sequence. Two call instances return values independent of each other. The first time an instance of this function is called, the

function returns a value of 1. Subsequent calls of the same instance return the previous value incremented by 1 (i.e., 2, 3, 4...).

Each time a data flow is called, all instances are reinitialized, starting at 1.

❖ Example

Use the function in a query's mapping expression to add a column of row IDs to a target.

```
gen_row_num()
```

9.3.53 gen_uuid

This function returns a unique identifier.

≡ Syntax

```
gen_uuid()
```

This function takes no parameters.

Return value

Returns a 32-character varchar string. For example, 550e8400e29b41d4a716446655440000.

9.3.54 get_domain_description

Returns the description of a value when given the domain name and the value.

≡ Syntax

```
get_domain_description(<domain_name>, <domain_value>)
```

Return value

varchar

The description is returned as a quoted string. If the value is not in the domain, then a NULL is returned.

Where

<code><domain_name></code>	Fully qualified domain name, including the database owner if required. For example: <code><datastorename> . <owner.domain></code> <code><datastorename> .. <domain></code>
<code><domain_value></code>	The constant value for which you want to return a description.

❖ Example

Function	Results
<code>get_domain_description('psoft..ACTION', 'ADL')</code>	"Additional"

9.3.55 get_env

Returns a value for the specified system environment variable.

≡ Syntax

```
get_env('<variable_name>')
```

Return Value

varchar(255)

Returns the value of the environment variable. Returns NULL if the environment variable is not set. You can use the `is_set_env` function to determine whether a variable is set.

Where

<code><variable_name></code>	The name of the environment variable. The name must be surrounded by single quotes.
------------------------------------	---

❖ Example

Function	Results
<code>getenv('TMP')</code>	<code>C:\Temp</code>

Related Information

[is_set_env](#) [page 1116]

9.3.56 get_error_filename

Returns the full path and file name for the error log, which Data Services generates after a job run. Data Services generates log files in the `<<DS_COMMON_DIR>>\log\<Job Server>\<repository>` directory. This log starts with 'error_'.

Data Services generates a different set of log files for each batch and real-time (in test mode) job run. Data Services creates only one set of log files during the life of a real-time service.

≡ Syntax

```
get_error_filename()
```

Return Value

varchar

❖ Example

Create a job and add a script with, for example, the following lines.

```
print('Error File Name:');  
print(get_error_filename());
```

Returns, for example, the following to the trace log:

```
Error File Name:  
d://DI11/log/JS_Ora/repo__riv/  
error_12_10_2004_12_06_41_12__8507da25_0b33_4ac1_9b53_fcf1e004c968.txt
```

9.3.57 get_file_attribute

Returns the value for a specified attribute of a physical file.

Syntax

```
get_file_attribute(<filename>, <attribute>)
```

Return value

double

If the attribute is size.

datetime

If the attribute is either date_created or date_modified.

Where

<filename>	The file name and path relative to the current Job Server. Enter a file name as a relative path or an absolute path.
<attribute>	One of the following attributes, which must be in single quotes: date_created, date_modified, or size.

Example

Function	Results
<pre>get_file_attribute('//order', 'date_created')</pre>	<pre>'2004:09:15:20:25:00'</pre> <p>The format in this example is YYYY:MM:DD:HH24:MM:SS.</p>
<pre>get_file_attribute(c:\database \order, 'size')</pre>	<pre>'63281'</pre> <p>Displays file size in bytes.</p>

Limitations

- The function is not pushed down.
- This function cannot be used in an ABAP data flow.

- For MS Window systems, this function cannot return the create time from FAT formatted disk drives. It works with the NTFS (New Technology File System) format. Most systems use NTFS today because it is more powerful and offers a security advantage over FAT.

9.3.58 get_monitor_filename

Returns the full path and file name for the monitor log, which Data Services generates during a job run. Data Services generates log files in the `<DS_COMMON_DIR>\log\<Job Server>\<repository>` directory. This log starts with 'monitor_'.

Data Services generates a different set of log files for each batch job run and each real-time job run (in test mode). Data Services creates only one set of log files during the life of a real-time service.

Syntax

```
get_monitor_filename()
```

Return Value

varchar

Example

Create a job and add a script with, for example, the following lines.

```
print('Monitor File Name');
print(get_monitor_filename());
```

Returns, for example, the following to the trace log:

```
Monitor File Name
d://DI11/log/JS Ora/repo_riv/
monitor_12_1_2004_12_06_41_12__8507da25_0b33_4ac1_9b53_fcfl1e004c968.txt
```

Note

A monitor log is referred to as a trace log in the Designer.

9.3.59 get_trace_filename

Returns the full path and file name for the trace log, which Data Services generates during a job run. Data Services generates log files in the `<DS_COMMON_DIR>\log\<Job Server>\<repository>` directory. This log starts with 'trace_'.

Data Services generates a different set of log files for each batch job run and each real-time job run (in test mode). Data Services creates only one set of log files during the life of a real-time service.

Syntax

```
get_trace_filename()
```

Return Value

varchar

Example

Create a job and add a script with, for example, the following lines.

```
print('Trace File Name');  
print(get_trace_filename());
```

Returns, for example, the following to the trace log:

```
Trace File Name  
d://DI11_97/log/JS_Ora/o920i1__riv/  
trace_12_10_2004_12_06_41_12__8507da25_0b33_4ac1_9b53_fcf1e004c968.txt
```

9.3.60 greatest

Returns the greatest of the list of one or more expressions.

Syntax

```
greatest(<expression_list>)
```

Return Value

Data Services uses the first expression to determine the return type. After comparison, the result is converted into the return data type.

Where

<expression_list>

A list of one or more comma-separated expressions.

Details

GREATEST returns the greatest of the list of one or more expressions. After comparison, the result is converted into a return data type. Data Services implicitly converts expression in the list to a normalized data type before comparison.

The following rules determine the normalized data type.

- If the return data type is varchar, then all expressions are implicitly normalized to varchar before comparison.
- If the return data type is one of the date data types, then all the expressions in the list are implicitly normalized to that data type before comparison. For example, if the return data type is date, and another data type is 'datetime', then the 'datetime' data type is normalized to 'date' before comparison.
- If the return data type is numeric, then all the expressions are implicitly normalized to the highest precedence numeric expression in the list. For example, greatest (expr1,expr2,expr3,expr4) where expr1 is as integer, expr2 is a decimal (4,2), expr3 is a float, expr4 is a decimal (38,7), then the normalized data type is decimal. All the expressions in the list are converted to decimal data type before comparison. If the normalized data type is decimal, then its precision is the highest precision among all decimal data type expressions. The decimal data type expressions preserve their scale during implicit conversion. When an integer data type expression is converted to a decimal data type, its scale is 0. When float, double and varchar data types are converted into decimal data types, their scale is 6.

Note

greatest() returns NULL when at least one argument is NULL.

❖ Example

Input				
ID	GRADE_Q1	GRADE_Q2	GRADE_Q3	GRADE_Q4
1	'A'	'B'	'B'	'C'
2	'F'	'F'	'E'	'C'
3	'B'	'B'	NULL	NULL
Output				
MAX_GRADE=greatest (GRADE_Q1 ,GRADE_Q2 ,GRADE_Q3 ,GRADE_Q4)				
ID	MAX_GRADE			
1	'C'			
2	'F'			

Output	
MAX_GRADE=greatest (GRADE_Q1, GRADE_Q2, GRADE_Q3, GRADE_Q4)	
ID	MAX_GRADE
3	NULL

9.3.61 host_name

Returns the name of the computer on which the job is executing.

≡ Syntax

```
host_name()
```

Return Value

varchar

⚙ Example

```
print('Host Name: [host_name()]');
```

9.3.62 ifthenelse

Allows conditional logic in expressions.

≡ Syntax

```
ifthenelse(<condition>, <true_branch>, <false_branch>)
```

Return value

<true_branch> or <false_branch>

Returns one of the values provided, based on the result of <condition>. The data type of the return value is the data type of the expression in <true_branch>. If the data type of <false_branch> is not convertible to

the data type of `<true_branch>`, Data Services produces an error at validation. If the data types are convertible but don't match, a warning appears at validation.

If `<condition>` compares a varchar value with trailing blanks, the `ifthenelse` function ignores the trailing blanks.

To compare a NULL value (NULL constant or variable that contains a NULL constant), use the IS NULL or IS NOT NULL operator. If you use the Equal (=) or Not equal to (< >) operator to compare against a NULL value, `<condition>` always evaluates to FALSE.

Where

<code><condition></code>	An expression that evaluates to TRUE or FALSE.
<code><true_branch></code>	An expression that the function returns if <code><condition></code> evaluates to TRUE.
<code><false_branch></code>	An expression that the function returns if <code><condition></code> evaluates to FALSE.

To improve performance, Data Services pushes this function to the database. Thus, the database evaluates the IFTHENELSE logic rather than the engine.

Use this function to apply conditional logic when mapping columns or selecting columns in a query. For more flexible control over conditions in a script, use the IF keyword in the scripting language.

❖ Example

Function	Results
<pre>ifthenelse (ZIP < 94000, 'SOUTH', 'NORTH')</pre>	If the value in the column ZIP is less than 94000, the value returned is SOUTH. If ZIP is greater than 94000, then the value returned is NORTH.

9.3.63 index

Returns the index of a given character sequence in a string.

≡ Syntax

```
index(<input_string>, <index_string>, <start>)
```

Return value

int

Specifies the first location of the indicated character sequence.

Where

<code><input_string></code>	The source string.
<code><index_string></code>	The character sequence sought in <code><input_string></code> .
<code><start></code>	The position where the function starts searching in <code><input_string></code> for the character sequence contained in <code><index_string></code> . <code><Start></code> should be a positive number between 1 and the length of <code><input_string></code> .

Details

The function searches for the `<index_string>` beginning at the `<start>` position in the `input_string`. If `<start>` is 0, it is reset to 1; if `<start>` is greater than the number of characters in `<input_string>`, the function returns NULL.

If `<index_string>` is not found in `<input_string>`, the function returns NULL. The characters in `<index_string>` must match exactly the sequence of characters in `<input_string>`. The search is case-sensitive.

❖ Example

Function	Results
<code>index('Accounting Department', 'DEPARTMENT', 1)</code>	0
<code>index('Accounting Department', 'Department', 1)</code>	12

9.3.64 init_cap

Changes the characters in a string to title case. This function converts the first letter of each word to uppercase and the rest of the value to lowercase. It ignores non-alphanumeric characters.

≡ Syntax

```
init_cap(<value>, '<locale>')
```

Return value

varchar

The title case string. Words are delimited by white space or characters that are not alphanumeric.

Where

<value>	The string to be modified.
<locale>	Optional parameter that converts the string to the specified locale.

i Note
ISO 639 language code and ISO 3166 country code formats are supported.

❖ Example

Function	Results
<code>init_cap('Data Services')</code>	'Data Services'
<code>init_cap(StreetAddress)</code>	Writes the value, for example '1234 west washington school road', in column <code>StreetAddress</code> as '1234 West Washington School Road'.
<code>Print(Init_cap('have a nice day -hyphen +plus _underscore \slash \$dollar *star @at tab mIXedWORD UPPER lower ! punctations 1234digits'));</code>	Have A Nice Day -Hyphen +Plus _Underscore \Slash \$Dollar *Star @At Tab Mixedword Upper Lower !Punctuations 1234digits
<code>init_cap(LastName, 'tr')</code>	The value in column <code>LastName</code> will have the first letter capitalized. If there is more than one last name in this column, that string will also have its first letter capitalized. It is also converted to the Turkish locale, using the ISO 639 language code.

Limitations

- The function is pushed down to Oracle databases only.
- You cannot use this function in an ABAP data flow.

Related Information

[ISO 639 language list](#)
[ISO 3166 Country Code list](#)

9.3.65 interval_to_char

Converts an interval value to a string.

Syntax

```
interval_to_char(<input_interval, interval_type>)
```

Return value

varchar
The converted string.

Where

<input_interval>	The value of type interval to convert.
<interval_type>	A string describing the format of the interval. Choose from the following values:
D	Days
H	Hours
M	Minutes
S	Seconds

Example

Function	Results
<pre>interval_to_char(start_date - sysdate(), 'd')</pre>	The number of days between the date in the column start_date and today's date.

Function	Results
<code>interval_to_char(start_time - systime(), 'm')</code>	The number of minutes between the time in the column <code>start_time</code> and the current time.

9.3.66 is_group_changed

Returns 1 if the group is changed, 0 otherwise.

Syntax

```
is_group_changed(<expression>)
```

Return Value

Integer

Where

`<expression>` One or more valid input expressions separated by commas.

Details

This function groups records based on the equal value of the input expressions in parameter1 in the natural order of the input record stream. It returns 1 when the group is changed, 0 otherwise. In the following example, the results show that four of the input groups have changed.

Example

Function	Results
<code>is_group_changed(state,city)</code>	1,0,1,0,0,1,1

Group ID	State	City	Group change
1	California	Los Angeles	1
2	California	Los Angeles	0
3	California	San Francisco	1
4	California	San Francisco	0
5	California	San Francisco	0
6	Nevada	Reno	1
7	Colorado	Reno	1

9.3.67 is_set_env

Verifies if the specified system environment variable is set.

Syntax

```
is_set_env(<variable_name>)
```

Return value

int

Returns 1 if the environment variable is set, otherwise, returns 0.

Where

<variable_name>

The name of the environment variable. The name must be surrounded by single quotes.

Example

Function

```
is_set_env('MODE')
```

Results

Returns 1 if the MODE variable has already been set; returns 0 if the MODE variable has not been set.

9.3.68 is_valid_date

Indicates if an expression can be converted into a valid calendar date value. For example the following will return a negative result:

```
is_valid_date ('01/34/2002', 'mm/dd/yyyy')
```

This expression returns 0 because there is no such date as January 34th.

Syntax

```
is_valid_date(<input_expression>,'<date_format>')
```

Return value

int

- If the expression is not NULL and valid, it returns 1.
- If the expression is not NULL and invalid, it returns 0.
- If the expression is NULL, it returns NULL.

Where

<code><input_expression></code>	The expression to be validated. If the expression does not resolve to a value of data type <code>varchar</code> , you will see a warning that the value has been converted to a <code>varchar</code> .
---------------------------------------	---

<code><date_format></code>	The string identifying the date format of the input string. Construct the date format using the following codes and other literal strings or punctuation:
----------------------------------	---

DD	2-digit day of the month
MM	2-digit month
MONTH	Full name of month
MON	3-character name of month
YY	2-digit year
YYYY	4-digit year

❖ Example

Function	Results
<code>is_valid_date (Orders.SubmitDate, 'mm/dd/yyyy')</code>	Tests whether the string <code>Orders.SubmitDate</code> can be converted to a calendar date with the <code>mm/dd/yyyy</code> date format.

Related Information

[date \[page 305\]](#)

9.3.69 is_valid_datetime

Indicates if an expression can be converted into valid calendar date and time values. For example the following will return a negative result:

```
is_valid_datetime ('01/14/2002 26:56:09', 'mm/dd/yyyy hh24:mi:ss')
```

This expression returns 0 because there is no such hour as "26", even on the 24 hour clock.

≡ Syntax

```
is_valid_datetime(<input_expression>, '<datetime_format>')
```

Return value

int

- If the expression is not NULL and valid, it returns 1.
- If the expression is not NULL and invalid, it returns 0.
- If the expression is NULL, it returns NULL.

Where

<code><input_expression></code>	The expression to be validated.
---------------------------------------	---------------------------------

`<datetime_format>` The string identifying the datetime format of the input expression. Construct the datetime format using the following codes and other literal strings or punctuation:

DD	2-digit day of the month
MM	2-digit month
MONTH	Full name of month
MON	3-character name of month
YY	2-digit year
YYYY	4-digit year
HH24	2-digit hour of the day (00-23)
MI	2-digit minute (00-59)
SS	2-digit second (00-59)

❖ Example

Function	Results
<pre>is_valid_datetime (Orders.Received, 'mm/dd/yyyy hh24:mi:ss')</pre>	Tests whether the string <code>Orders.Received</code> can be converted to the <code>mm/dd/yyyy hh24:mi:ss</code> datetime format.

Related Information

[datetime](#) [page 307]

9.3.70 is_valid_decimal

Indicates if an expression can be converted into a valid decimal value.

≡ Syntax

```
is_valid_decimal(<input_expression>, '<decimal_format>')
```

Return value

int

- If the expression is not NULL and valid, it returns 1.
- If the expression is not NULL and invalid, it returns 0.
- If the expression is NULL, it returns NULL.

Where

<input_expression>	The expression to be validated.
<decimal_format>	<p>A string indicating the decimal format of the input expression. Use pound characters (#) to indicate digits and a decimal indicator. If necessary, include commas as thousands indicators. For example, to specify a decimal format for numbers smaller than 1 million with 2 decimal digits, use the following string: '#,###,###.##'.</p> <p>To indicate a negative decimal number, add a minus "-" sign at the beginning or end of this value. For example, to test if the stock price difference can be converted to decimal format, use the following function:</p> <pre>is_valid_decimal (Stocks.Price_difference, '-###.##')</pre>

❖ Example

Function	Results
<code>is_valid_decimal (Orders.Price, '###,###.##')</code>	Tests whether the string <code>Orders.Price</code> can be converted to decimal format.

9.3.71 is_valid_double

Indicates if an expression can be converted into a valid double value.

≡ Syntax

```
is_valid_double(<input_expression>, '<double_format>')
```

Return value

int

- If the expression is not NULL and valid, it returns 1.
- If the expression is not NULL and invalid, it returns 0.

- If the expression is NULL, it returns NULL.

Where

<code><input_expression></code>	The expression to be validated.
<code><double_format></code>	A string indicating the double format of the input expression. Use pound characters (#) to indicate digits and a decimal indicator. If necessary, include commas as thousands indicators. For example, to specify a double format for numbers smaller than 1 million with 2 decimal digits, use the following string: <code>#,###,###.##</code>

Example

Function	Results
<code>is_valid_double (Product.Weight, '###.###')</code>	Tests whether the string <code>Product.Weight</code> can be converted to double format.

9.3.72 is_valid_int

Indicates if an expression can be converted into a valid integer value.

Syntax

```
is_valid_int(<input_expression>, '<int_format>')
```

Return value

int

- If the expression is not NULL and valid, it returns 1.
- If the expression is not NULL and invalid, it returns 0.
- If the expression is NULL, it returns NULL.

Where

<code><input_expression></code>	The expression to be validated.
<code><int_format></code>	The format specifying the thousands separator of the input expression. For example, to specify an integer format, use the following string: <code>###.###</code> . Valid separators include the period (.) and the comma (,). However, you can only use one valid separator type in a format. Separator defaults to the comma (,) when none is specified.

❖ Example

Function	Results
<pre>is_valid_int (QuarterResults.Volume, '###.###')</pre>	Tests whether the string <code>QuarterResults.Volume</code> can be converted to the <code>###.###</code> integer format.

9.3.73 is_valid_real

Indicates if an expression can be converted into a valid real value.

≡ Syntax

```
is_valid_real(<input_expression>, '<real_format>')
```

Return value

int

- If the expression is not NULL and valid, it returns 1.
- If the expression is not NULL and invalid, it returns 0.
- If the expression is NULL, it returns NULL.

Where

<code><input_expression></code>	The expression to be validated.
---------------------------------------	---------------------------------

<code><real_format></code>	A string indicating the real format of the input expression. Use pound characters (#) to indicate digits and a decimal indicator. For example, to specify a real format for numbers smaller than 1 million with 2 decimal digits, use the following string: '#,###,###.##'.
----------------------------------	---

Example

Function	Results
<code>is_valid_real (QuarterResults.Mean, '#,###.#####')</code>	Tests whether the string <code>QuarterResults.Mean</code> can be converted to real format.

9.3.74 is_valid_time

Indicates if an expression can be converted into a valid time value.

Syntax

```
is_valid_time(<input_expression>,'<time_format>')
```

Return value

int

- If the expression is not NULL and valid, it returns 1.
- If the expression is not NULL and invalid, it returns 0.
- If the expression is NULL, it returns NULL.

Where

<code><input_expression></code>	The expression to be validated.
<code><time_format></code>	The string identifying the time format of the input expression. Construct the time format using the following codes and other literal strings or punctuation:

HH24	2-digit hour of the day (00-23)
MI	2-digit minute (00-59)
SS	2-digit second (00-59)

❖ Example

Function	Results
<code>is_valid_time (Orders.ReceivedTime, 'hh24:mi:ss')</code>	Tests whether the string <code>Orders.ReceivedTime</code> can be converted to the <code>hh24:mi:ss</code> datetime format.

9.3.75 isempty

Indicates if a nested table contains data.

≡ Syntax

```
isempty(<table_name>)
```

Return value

int

The result of the content test: returns 1 if the table does not contain data; returns 0 if the table does contain data.

Where

`<table_name>`

The fully qualified name of the nested table to test. A fully qualified name contains the parent table names up to the top level of the table in the current context.

If you only specify a table name, Data Services looks for the table among the tables available through the FROM clause of the current context. If you specify a partially qualified table name (only part of the table hierarchy), Data Services looks for the table among the tables available in the FROM clause of the context indicated by the partial qualification.

When performing operations on hierarchical data, the `isempty` function allows you to exclude rows in a higher-level table based on whether a lower-level table contains data.

Data Services determines that a nested table is empty when the table contains no rows. If the nested table contains even one row with null values in all columns, the `isempty` function indicates that the table has content. If the nested table is empty except for another nested table, and the second nested table does contain data, then the first nested table is not empty.

❖ Example

You can use the `isempty` function to determine if there are line items associated with a sales order. For example, if the sales order is the input data set to a Query transform and you want the query to exclude orders without line items, include the following expression in the WHERE clause of the top-level context of the query:

```
isempty (order_table.line_items_table)
```

9.3.76 isweekend

Indicates if a date corresponds to Saturday or Sunday.

≡ Syntax

```
isweekend (<date1>)
```

Return value

int

The result of the date test: returns 1 if the date is a Saturday or Sunday; returns 0 if not.

Where

<date1>

The value of type date or datetime to test.

❖ Example

Function	Results
<code>isweekend(hire_date)</code>	Tests whether the date in <code>hire_date</code> is a Saturday or Sunday.
<code>isweekend(SYSDATE)</code>	Tests whether today is a Saturday or Sunday.

9.3.77 job_name

Returns the name of the job in which the call to this function exists.

≡ Syntax

```
job_name()
```

Return Value

varchar

❖ Example

```
print('Starting execution of Job: [job_name()] as user:  
[system_user_name()]');
```

9.3.78 Job_Run_ID

≡ Syntax

Retrieves the job run ID for the current job execution.

Syntax

```
job_run_id()
```

Return value

Varchar (20)

9.3.79 julian

Converts a date to its integer Julian value, the number of days between the start of the Julian calendar and the date.

Syntax

```
julian(<date1>)
```

Return value

int: The Julian representation of the date.

Where

<date1>

The source value of type date or datetime.

Example

Function	Results
<pre>julian(to_date('Apr 19, 1997', 'mon dd, yyyy'))</pre>	729436

9.3.80 julian_to_date

Converts a Julian value to a date.

Syntax

```
julian_to_date(<input_julian>)
```

Return value

date

The date that corresponds to the input Julian value.

Where

<input_julian>

An integer representing the Julian value to be converted.

Example

Function	Results
<pre>julian_to_date(Julian_Date)</pre>	Converts the number indicated by Julian_Date to its date value.

9.3.81 key_generation

Generates the next value in a series, after determining the last value in the series.

The key_generation function determines the maximum existing key value in a given column in the table in the database manager and uses that value as a starting point to generate key values for the target schema.

Note

The key_generation function is not supported when using SAP tables as readers.

Syntax

```
key_generation (<table>, <key_column>, <key_increment>)
```

Return value

int

The column value found to meet the function requirements.

Where

<code><table></code>	<p>The full path name of the file or full database specification of the table in which the <code><key_column></code> is located. Enclose this value in single quotation marks.</p> <p>Use a fully qualified table name that includes the datastore, owner, and table name. For example: <code>oracle_ds.TIGER.sales</code>.</p> <p>For Netezza 7.x multi-schema, include the datastore, owner, schema and table name. For example: <code>NZ7_ds.DSDEV.SCHEMA1.TABLE1</code>. If the schema name <code>SCHEMA1</code> is not specified, the default schema is used. For example: <code>NZ7_ds.DSDEV.TABLE1</code>.</p>
<code><key_column></code>	<p>A column with existing keys from which this function determines the largest existing key value. Enclose this value in single quotation marks.</p>
<code><key_increment></code>	<p>The integer increment used between key values this function generates.</p>

❖ Example

Function	Results
<code>key_generation('target_ds.dbo.sales', 'order_number', 1)</code>	Looks for the last key value in the <code>order_number</code> column in the sales database and returns the largest value plus one.

9.3.82 last_date

Returns the last date of the month for a given date.

≡ Syntax

```
last_date(<in_date>)
```

Return Value

date

Where

<code><in_date></code>	The date for which the last date of the month is to be calculated.
------------------------------	--

❖ Example

Function	Returns
<code>last_date('1990.10.01')</code>	<code>'1990.10.31'</code>

9.3.83 least

Returns the least of the list of one or more expressions.

≡ Syntax

```
least(<expression_list>)
```

Return Value

Data Services uses the first expression to determine the return type. After comparison, the result is converted into the return data type.

Where

<code><expression_list></code>	A list of one or more comma-separated expressions.
--------------------------------------	--

Details

least returns the least of the list of one or more expressions. After comparison, the result is converted into a return data type. Data Services implicitly converts expressions in the list to a normalized data type before comparison.

The following rules determine the normalized data type:

1. If the return data type is varchar, then all expressions are implicitly normalized to varchar before comparison.
2. If the return data type is one of the date data types, then all the expressions in the list are implicitly normalized to that data type before comparison. For example, if the return data type is date, and another data type is 'datetime', then the 'datetime' data type is normalized to 'date' before comparison.
3. If the return data type is numeric, then all the expressions are implicitly normalized to the lowest precedence numeric expression in the list. For example, least(expr1,expr2,expr3,expr4) where expr1 is an integer, expr2 is a decimal (4,2), expr3 is a float, expr4 is a decimal (38,7), then the normalized data type is decimal. All the expressions in the list are converted to decimal data type before comparison. If the normalized data type is decimal, then its precision is the lowest precision among all decimal data type expressions. The decimal data type expressions preserve their scale during implicit conversion. When an integer data type expression is converted to a decimal data type, its scale is 0. When float, double and varchar data types are converted into decimal data types, their scale is 6.

Note

least() returns NULL when at least one argument is NULL.

❖ Example

Input				
ID	GRADE_Q1	GRADE_Q1	GRADE_Q3	GRADE_Q4
1	'A'	'B'	'B'	'C'
2	'F'	'F'	'E'	'C'
3	'B'	'B'	NULL	NULL

Output		
MIN_GRADE=least (GRADE_Q1, GRADE_Q2, GRADE_Q3, GRADE_Q4)		
ID	MAX_GRADE	MIN_GRADE
1	'C'	'A'
2	'F'	'C'
3	NULL	NULL

9.3.84 length

Returns the number of characters in a given string.

Syntax

```
length(<value>)
```

Return value

integer

The number of characters in <value>.

Where

<value>

A string indicating the column name, variable, or other element whose length is calculated.

❖ Example

In the [Mapping](#) box of a query, you can use the length function to return the number of characters in each row of a column. For example, with the OUTPUT field selected in the target schema of a query, entering the following statement in the [Mapping](#) box:

```
length(dal_emp.ename)
```

produces the following results:

Source column (dal_emp.ename)	Target column (output)
jones	5
nguyen	6
tanaka	6

9.3.85 literal

Returns an input constant expression without interpolation. Data Services normally does not use variable interpolation on constants. However, if you pass in a variable as a constant expression, Data Services automatically uses variable interpolation, replacing special characters.

This is an issue with the `Match_pattern` and `Match_regex` functions because they require these special characters. If your `pattern_string` or `regular_expression_pattern` parameter in these functions is a constant, you may want to disable interpolation. If so, use the `Literal` function.

If, for example, you want to match `$my_pattern` with the pattern `'PART[123]'`.

If you coded it simply as:

`$my_pattern = 'PART[123]'; match_pattern(product, $my_pattern);` the interpolation would actually change the pattern being matched to `'PART123'`, but if you code it as:

```
$my_pattern = literal ('PART[123]');  
match_pattern(product, $my_pattern);
```

then it could return 1, because the pattern would remain `'PART[123]'`.

Alternatively, if you do not want to use a variable, you can code it as:

```
match_pattern (product, 'PART[123]');
```

because in this case no interpolation is done on the constant `'PART[123]'`.

There is no runtime cost for the `Literal` function. Data Services substitutes the constant expression at compile time.

Syntax

```
literal(<input>)
```

Return value

Same as that of the value given for the input parameter but without interpolation.

Where

<input>

A constant expression of any data type.

Example

If you want to match only `PART1` or `PART2` or `PART3` using the `Match_pattern` function, you must assign a pattern to a variable without interpolation. Use the `Literal` function in the following type of expression:

```
$pattern = literal('PART[123]');
```

For example, if you do not use the `literal` function, the value assigned to `$my_pattern` in the following sample is `'PART123'` because Data Services automatically removes square brackets during interpolation.

```
$my_pattern='PART[123]';  
print($my_pattern);  
if (match_pattern('PART1',$my_pattern) <> 0)  
  print('Matched');
```

```
else
print('Not Matched');
```

To disable interpolation, use the Literal function. The following example returns the result you want.

```
$my_pattern=LITERAL('PART[123]');
print($my_pattern);
if (match_pattern('PART1',$my_pattern) <> 0)
print('Matched');
else
print('Not Matched');
```

9.3.86 ln

Returns the natural logarithm of the given numeric expression.

⌘ Syntax

```
ln(<numeric_expression>)
```

Return Value

Float

Where

<numeric_expression>

Any numeric expression.

Details

Returns the natural logarithm of the given numeric expression. Return value is NULL if input is negative.

⚙ Example

Function	Results
ln(5.436563656918)	1.693147

9.3.87 load_from_gcs_to_gbq

a function that uses information from the named file location object to copy data from Google Cloud Storage into Google BigQuery tables.

Use this function in a workflow script to transfer data from a Google Cloud Storage into Google BigQuery tables to be used as a source in a data flow. The software uses the local and remote paths and Google Cloud Storage protocol information from the named file location object.

Syntax

```
load_from_gcs_to_gbq("<datastore_name>", "<remote_file_name>",  
"<table_name>", "<write_mode>", "<file_format>")
```

Return value

int

Returns 1 if function is successful. Returns 0 if function is not successful.

Where

<code><datastore_name></code>	Name of the Google BigQuery datastore.
<code><remote_file_name></code>	Name of the file to copy from the remote server in the format <code>gs://bucket/filename</code> . Wildcards may be used.
<code><table_name></code>	Name of the Google BigQuery table name in the format <code>dataset.table</code> .
<code><write_mode></code>	(Optional.) The write mode value can be append (default) or truncated.
<code><file_format></code>	The format of the data files using one of the following values: <ul style="list-style-type: none">• CSV: For CSV files. This is the default value.• DATASTORE_BACKUP: For datastore backups.• NEWLINE_DELIMITED_JSON: For newline-delimited JSON.• AVRO: For Avro.

❖ Example

To copy a file `json08_from_gbq.json` from a Google BigQuery datastore named `NewGBQ1` on a remote server to a Google BigQuery table named `test.json08` on a local server, set up a script object that contains the `load_from_gcs_to_gbq` function as follows:

Sample Code

```
load_from_gcs_to_gbq('NewGBQ1', 'gs://test-bucket_1229/from_gbq/
json08_from_gbq.json', 'test.json08', 'append', 'NEWLINE_DELIMITED_JSON');
```

9.3.88 load_from_s3_to_redshift

Uses the Redshift `COPY` command to copy data files from an Amazon Simple Storage Service (S3) bucket to a Redshift table.

Before using this function, set up an S3 file location object. For more information, see [Amazon S3 protocol options \[page 176\]](#).

Syntax

```
load_from_s3_to_redshift("<datastore name>", "<table name>", "<file location
name>", "<file name>", "<file options>")
```

Where

<code><datastore name></code>	Name of the Redshift datastore.
<code><table name></code>	<p>Name of the target table.</p> <p>You can also specify the following:</p> <ul style="list-style-type: none">• <code><table name> = <table name></code>• <code><table name> = <schema name>.<table name></code>• <code><table name> = <Redshift datastore name>.<schema name>.<table name></code>• <code><table name> = <Redshift datastore name>.<Alias name used in datastore>.<table name></code>
<code><file location name></code>	Location of the S3 file.
<code><file name></code>	Fully qualified name of the Amazon S3 file to copy to the Redshift table. Wild cards are allowed.

<file options>

(optional) Use the following file options as applicable when copying a file:

- *acceptanydate*: Accepts any date, even those with invalid formats, without throwing an error.
- *acceptinvchars*: Replaces invalid UTF-8 characters.
- *blankasnull*: Inserts null if the input data is blank.
- *dateformat*: Defines the date format. For example, 'YYYY-MM-DD'.
- *delimiter*: Defines the column delimiter. For example, '|'.
- *emptyasnull*: Inserts null if input data is empty.
- *encoding*: Defines the data file encoding type. Valid values include utf8 (default), utf16, utf16le, and utf16be.
- *encrypted*: Loads encrypted data files from S3.
- *escape*: Removes escape (\) character. For example, a\b\c would be a\b\c.
- *explicit_ids*: Data values must match the Identity format and Identity columns.
- *fillrecord*: Fills null if any record is missed.
- *ignoreblanklines*: Ignores blank lines.
- *ignoreheader*: Skips the specified number rows as a file header. The default is 0.
- *manifest*: Loads manifest data files from S3.
- *maxerror*: Defines the maximum number of errors allowed. The default is 0.
- *null as*: Defines the special null string
- *removequotes*: Removes quotes from the data file.
- *roundec*: Rounds up numeric values when the input value is greater than the scale defined for the column.
- *timeformat*: Defines the timestamp format. For example, 'YYYY-MM-DD HH:MI:SS'.
- *trimblanks*: Removes whitespace characters. Only applies to the varchar data type.
- *truncatecolumns*: Truncates data in columns when the input value is greater than the column defined. Applies to varchar or char data types and rows 4MB or less in size.
- *gzip*: Loads compressed data files from S3.
- *lzop*: Loads compressed data files from S3.
- *bzip2*: Loads compressed data files from S3.

Sample Code

```
CREATE __AL_REPO_FUNCTION load_from_s3_to_redshift("Datastore" __FUNC_CHAR
IN, "Table name" __FUNC_CHAR IN, "File location name" __FUNC_CHAR IN,
"File name" __FUNC_CHAR IN, "File options" __FUNC_CHAR IN )
SET(database_type = 'ACTA',
function_type = 'Miscellaneous Function',
DB_FunctionName = 'load_from_s3_to_redshift',
Description = 'This function loads Amazon S3 data file(s) to a Amazon
Redshift table',
Parallelizable = 'NO',
External_name = 'load_from_s3_to_redshift',
return_param_dep = 'null',
return_datatype = '5',
return_datatype_size = '4',
param0 = 'Name of the Amazon Redshift datastore.',
param1 = 'Name of the target table.',
param2 = 'Name of the Amazon S3 File location.',
param3 = 'Fully qualified name of the Amazon S3 file(s). Wild cards are
allowed.',
param4 = 'File options that can be applied when copying the file. For
example, \'delimiter \',\' encoding \'utf8\'\'.'
)
```

❖ Example

To copy a data file inside `<bucket name>/<sub directory>` on S3 to a Redshift table, define the following in the S3 datastore:

- Bucket = `<bucket name>`
- Remote Directory = `<sub directory>`

Then enter the following:

```
load_from_s3_to_redshift('redshift_ft', 'customer ', 'S3_to_Redshift',
'customer.dat', 'delimiter \',\' ');
```

❖ Example

To generate an AES256 key, enter the following:

```
encrypt_aes('<plain password>', '<passphrase>', 256)
```

You can then use the key to upload data from the Redshift table to the S3 bucket.

```
unload ('select * from <redshift table>')
to 's3://<bucket name>/<sub directory>/'
credentials 'aws_access_key_id=<access key>;aws_secret_access_key=<secret
access key>;master_symmetric_key=<AES256 key>'
delimiter '|' encrypted bzip2;
```

To copy the encrypted data files on S3 back to a Redshift table, enter the following:

```
load_from_s3_to_redshift('redshift_ft', 'public.t31_household',
'S3_to_Redshift_3', 't31_encrypted', 'master_symmetric_key \'<AES256 key>' \
encrypted bzip2 delimiter '\|\'');
```

❖ Example

To copy JSON data from S3 to a Redshift table, with a JSON path, enter the following:

```
load_from_s3_to_redshift('redshift_ft', 'public.t32_category',
'S3_to_Redshift_3', 't33_category.json', 'json \'s3://dsqa-redshift-bkt3/
t33_category_jsonpath.json\');
```

❖ Example

To copy CSV data from S3 to a Redshift table, enter the following:

```
load_from_s3_to_redshift('redshift_ft', 'public.t32_category',
'S3_to_Redshift_3', 't34_category_csv.txt', 'csv quote as \'%\';
```

❖ Example

To copy fixed-width data from S3 to a Redshift table, enter the following:

```
load_from_s3_to_redshift('redshift_ft', 'public.t35_fixed_width',
'S3_to_Redshift_3', 't35', 'fixedwidth \'catid:5,catgroup:10,catname:
9,catdesc:40\');
```

9.3.89 load_to_xml

This function converts NRDM into XML and places it in a single column during a data load.

If the function fails due to an error when trying to produce the XML output, Data Services returns NULL for scalar columns (that you select for output) and an empty nested table for NRDM columns.

Syntax

```
load_to_xml (nested_table_name, schema_dtd_name, enable_validation,  
xml_header, replace_string_nulls, is_top_level_element, max_size)
```

Where

nested_table_name	The name of the NRDM table that you want to convert into an XML format.
schema_dtd_name	<p>The name of the DTD or XML Schema that you want Data Services to use to format the XML text in the output column.</p> <p>Be sure to match the structure of the nested table to that provided by the DTD or XML Schema (or match the structure of its only child). Otherwise, Data Services will not produce output. See the <code>is_top_level_element</code> for an example of specifying an only child.</p>
	<div>Note You can generate an XML Schema or DTD from an NRDM schema using Data Services.</div>
enable_validation	Enter 1 to validate. Enter 0 to disable the validation parameter. Validates that Data Services generates XML that matches the XML format you specify for <code>schema_dtd_name</code> .
xml_header	<p>If the value specified is NULL, the header of the output XML has UTF-8 in the header. The default header generated is:</p> <pre><?xml version="1.0" encoding="UTF-8"?></pre> <p>If this value is not null, Data Services will replace the default XML header with the one you provide. Make sure the XML_header you provide matches the encoding of the target datastore where you will store the XML data.</p>
replace_null_string	If this value is not null, Data Services replaces NULL values with the specified string.

<code>is_top_level_element</code>	<p>Enter 1 if the input column representing the NRDM table matches the root element of the given DTD or XML Schema. Enter 0, if the input column should match the only child of the root element of the DTD or XML Schema.</p> <p>Examples:</p> <ul style="list-style-type: none"> Imagine that an XML reader creates an NRDM structure with the root element named PO and an XML Schema po.xsd, which defines a root element also called PO. In this case, when you select the input column name PO as the <code>nested_table_name</code>, it matches the root element of the schema, so set the value of <code>is_top_level_element</code> to 1. By contrast, imagine that a database table called employees has rows which each contain information about one employee. Also, you are using an XML Schema called employees.xsd. This schema defines a root element called allEmployees and defines a single element called employee (with unbounded occurrence). In Data Services, the employee table exists in an NRDM structure with employee as the root name. If you enter employee as the <code>nested_table_name</code>, because it does not match the root element of the schema allEmployees, set the value for <code>is_top_level_element</code> to 0. The function returns data with allEmployees as the root element and the number of employee elements under it to match the number of rows in the input employees table.
<code>max_size</code>	The expected maximum size of the generated XML.

❖ Example

```
load_to_xml(nested_table_name, billing_address_schema, 0, '<?xml
version="1.0" encoding="UTF-8"?>', NULL, 1, 4000)
```

9.3.90 local_to_utc

Converts the input datetime of any time zone to Coordinated Universal Time (UTC).

≡ Syntax

```
local_to_utc(<input datetime>, <timezone of the input with UTC offset>)
```

Return Value

datetime

Details

Converts the input datetime of any time zone to Coordinated Universal Time (UTC). The second parameter UTC offset is a constant value. If the UTC offset is not provided, then it is taken as the time zone of the job server host to calculate the UTC offset.

❖ Example

Function	Results
<code>local_to_utc('2014.02.01 00:00:00', 'UTC+08:30')</code>	'2014.01.31 15:30:00'

9.3.91 log

Returns the base-10 logarithm of the given numeric expression.

≡ Syntax

```
log(<num>)
```

Return Value

Float

Where

<num>

The number for which you want a base- 10 logarithm returned.

Details

Returns the base-10 logarithm of the given numeric expression. Return values is NULL if input is negative.

❖ Example

Function	Results
<code>log(100.000)</code>	2.000000

9.3.92 long_to_varchar

Converts a data type value of a given column from long to varchar.

≡ Syntax

```
long_to_varchar(<column_name>, <max_size>, start_position)
```

Return value

varchar

Where

<code><column_name></code>	The name of the column containing the long data type.
<code><max_size></code>	The maximum size for the converted data in the table column.
<code><start_position></code> (Optional)	Starting character position from which data is converted. The default start position is 1. A negative number indicates that the starting position is counted backwards from the last character.

❖ Example

```
long_to_varchar(content_column, 4000)
long_to_varchar(content_column, 4000, -5000)
```

9.3.93 lookup

Retrieves a value in a table or file based on the values in a different source table or file.

i Note

You cannot use this function with a J. D. Edwards datastore. Use the `lookup_ext` function instead.

≡ Syntax

```
lookup (<lookup_table>, <result_column>, <default_value>, <cache_spec>,
<compare_column>, <expression>)
```

Return value

any type

The value in the `<lookup_table>` that meets the lookup requirements. The return type is the same as `<result_column>`.

Where

<code><lookup_table></code>	<p>The table or file that contains the result or value you are looking up (<code><result_column></code>). The <code><compare_column></code> is also located in this table.</p> <p>Use a fully qualified table name that includes the datastore, owner, and table name. For example: <code>oracle_ds.TIGER.sales</code>.</p> <p>For Netezza 7.x multi-schema, include the datastore, owner, schema and table name. For example: <code>NZ7_ds.DSDEV.SCHEMA1.TABLE1</code>. If the schema name <code>SCHEMA1</code> is not specified, the default schema is used. For example: <code>NZ7_ds.DSDEV.TABLE1</code>.</p> <p>You might need to put the owner in quotes, particularly if you use lower case letters.</p>
<code><result_column></code>	<p>The column containing the values you want to retrieve. This column is in the <code><lookup_table></code>.</p>
<code><default_value></code>	<p>The value returned when there is no matching row in the <code><lookup_table></code>.</p>

<code><cache_spec></code>	<p>The caching method the lookup operation uses. List within single quotes. There are three possible settings:</p> <p>NO_CACHE: Reads values from the <code><lookup_table></code> for every row without caching values.</p> <p>PRE_LOAD_CACHE: Loads the <code><result_column></code> and <code><compare_column></code> into memory after applying filters but before executing the function.</p> <p>Select this option if the number of rows in the table is small or you expect to access a high percentage of the table values.</p> <p>DEMAND_LOAD_CACHE: Loads <code><result_column></code> and <code><compare_column></code> values into memory as the function identifies them.</p> <p>Select this option if the number of rows in the table is large and you expect to access a low percentage of the table values frequently.</p> <p>Select this option when you use the table in multiple lookups and the compare conditions are highly selective, resulting in a small subset of data.</p>
<code><compare_column></code>	<p>The column in the <code><lookup_table></code> that the function uses to find a matching row. When the function reads a varchar column in the <code><lookup_table></code>, it does not trim trailing blanks.</p>
<code><expression></code>	<p>The value that the function searches for in the <code><compare_column></code>. This can be a simple column reference, such as a column found in both a source and the <code><lookup_table></code>. This can also be a complex expression given in terms of constants and input column references.</p> <p>When <code><expression></code> refers to a unique source column, you do not need to include a table name qualifier. If <code><expression></code> is from another table or is not unique among the source columns, you need a table name qualifier.</p> <p>If <code><expression></code> is an empty string, the function searches for a zero-length varchar value in the <code><compare_column></code>.</p> <p>The function ignores trailing blanks in comparisons of <code><expression ></code> and values in <code><compare_column></code>.</p>

Note

You can specify more than one `<compare_column>` and `<expression>` pair—simply add additional pairs at the end of the function statement. The values must match for all specified pairs in order for the lookup function to find a matching row.

The lookup function uses a value you provide (`<expression>`) to find a corresponding value in a file or different table. Specifically, the function searches for the row in the `<lookup_table>` where the value in the `<compare_column>` matches the value in `<expression>`. The function returns the `<result_column>` value from this matching row.

For example, if your source schema uses a customer ID to identify each row, but you want the customer name in your target schema, you can use the lookup function to return the customer name given the customer ID.

In SQL terms, the lookup function evaluates `<expression>` for each row, then executes the following command:

```
SELECT <result_column>
```

```
FROM <lookup_table>
WHERE <compare_column> = <expression>
```

The value returned by this SELECT statement is the result of the lookup function for the row.

You can specify multiple `<compare_column>` and `<expression>` pairs to uniquely identify the `<result_column>` value. However, the software only provides fields for one pair; add extra `<compare_column>` and `<expression>` pairs to the output.

When there are no matching rows in the `<lookup_table>`, the lookup function returns the `<default_value>`. When multiple matching rows exist in the `<lookup_table>`, the row returned is based on whether the lookup table is a standard RDBMS table, an SAP application table, or a flat file:

- For standard RDBMS tables, the lookup function will find the matching row with the maximum value in the `<result_column>` and return that value.
- For SAP application tables or flat files, the lookup function randomly selects a matching row and returns the value in the `<result_column>` for that row.

Note

To avoid random row selection when the `<lookup_table>` is an SAP application table or a flat file, it is recommended that you use the `lookup_seq()` function.

To enhance performance, you can configure the lookup function to hold the values from the `<lookup_table>` in memory. To do so, use the `<cache_spec>` setting. The optimal setting depends on the number of rows the function must read, the number of rows in the table, and the available memory.

❖ Example

You can use the lookup function to return a text value given a numerical identifier. For example, suppose you have a source table containing a numerical identifier, such as an employee number, and you want to use the employee's name in your target.

You can use the lookup function to return the employee name based on the employee number. The lookup function uses a third table that translates the values from the source table to the desired values in the target table.

To produce the desired target column, select the column that you want to look up in the target schema. Click the [Functions](#) button, located over the [Mapping](#) text box. The function wizard opens. Under [Function categories](#), select [Miscellaneous_Function](#), then under [Function name](#), select [lookup](#). Click [Next](#). The Define Input Parameters window opens.

Enter the function parameters as follows:

Option	Value
Lookup table	ODS_DS.SSB.EMPLOYEE
Result column	LastName
Default value	'NoLastName'
Cache spec	'No_CACHE'
Compare column	EmployID
Expression	EmployID

The `<expression>` value refers to a column in the source file or table and therefore does not require qualification with a table name. If this `<expression>` was from another table or was not unique among the source columns, it would require a table name qualifier.

The function wizard automatically produces the mapping text.

```
lookup(SQL_rbh.DBO.ODS_EMPLOYEE, LastName, 'NoLastName', 'NO_CACHE',  
      EMP_ID, EMP_ID)
```

You can create a lookup function with two `<expression>` and `<compare_column>` pairs:

```
lookup(sap_ds.VBUP, GBSTA, 'none', 'NO_CACHE', VBELN, VBAK.VBELN, POSNR,  
      VBAP.POSNR)
```

This function returns the value from the GBSTA column in the VBUP table that corresponds to the VBELN value in the VBAK table and the POSNR value in the VBAP table. When no corresponding value is found, the function returns "none."

9.3.94 lookup_ext

≡ Syntax

The following syntax includes line breaks for clarity.

i Note

This function has a graphical editor.

```
lookup_ext(  
  [<lookup_table,cache_spec,return_policy>],  
  [<return_column_list>],  
  [<default_value_list>],  
  [<condition_list>],  
  [<orderby_column_list>],  
  [<output_variable_list>],  
  [<sql_override>]  
  set ("run_as_separate_process"='yes'),  
  ("output_cols_info"='<?xml version="1.0"  
encoding="UTF-8"?>  
<output_cols_info>  
<col index="1" expression="yes"/>  
</output_cols_info>')  
)
```

Return value

any type

The return type is the first lookup column in `<return_column_list>`.

Where

`<lookup_table>`

The table, file, or memory datastore that contains the result(s) or value(s) you are looking up (`<result_column_list>`). If the `<lookup_table>` is a database table, use the `<datastore.owner.table>` format. For example: ERP_ds.OWNER.EMPLOYEES

For Netezza 7.x multi-schema, include the datastore, owner, schema and table name. For example: NZ7_ds.DSDEV.SCHEMA1.TABLE1. If the schema name SCHEMA1 is not specified, the default schema is used. For example: NZ7_ds.DSDEV.TABLE1.

If the `<lookup_table>` is a flat file, use the `<file_ds.filename>` format. For example: delim."c:/temp/employees"

To substitute a variable for a file name, replace the data inside the double quotes, for example delim. "\$employees". The variable used to store a file name can be a local or global variable or a parameter passed to a work flow or a data flow. If the cache specification is NO_CACHE, SAP Data Services can pass in a different file name each time it calls lookup_ext. For example, you can call lookup_ext in a WHILE loop and assign a different file name to the variable passed as the lookup file. If the cache specification is PRE_LOAD_CACHE or DEMAND_LOAD_CACHE, only the first file name passed is used. The software ignores all file names passed during subsequent calls.

If the `<lookup_table>` is a memory datastore table, use the `<memory_ds.table>` format. For example: mem_ds.employees

`<cache_spec>`

The caching method the lookup_ext operation uses. You can select one of the following cache specifications:

- NO_CACHE—Reads values from the `<lookup_table>` for every row without caching values.
- PRE_LOAD_CACHE—Loads the `<return_column_list>`, `<compare_column>` (see `<condition_list>`), and `<orderby_column_list>` into memory after applying constant filters and before executing the function.
Select this option if the number of rows in the table is small or you expect to access a high percentage of the table values.
- DEMAND_LOAD_CACHE—Loads `<return_column_list>`, `<compare_column>` (see `<condition_list>`), and `<orderby_column_list>` into memory as the function identifies them.
Select this option if the number of rows in the table is large and you expect to frequently access a low percentage of table values.
Select this option when you use the table in multiple lookups and the compare conditions are highly selective, resulting in a small subset of data.

`<return_policy>`

Use `<return_policy>` when you expect multiple rows and want output data from one of the selected rows. This optional parameter specifies whether the return columns should be obtained from the smallest or the largest row based on values in the order by columns. The value can be MAX (default), MAX-NS, MIN, or MIN-NS. MAX-NS and MIN-NS allow the lookup_ext function to treat NULL as the smallest value instead of the largest value.

<return_column_list>	<p>A comma-separated list containing the names of output columns in the <lookup_table>.</p> <p>For a given output column in the lookup table, select the Expression? check box in the lookup_ext editor when some of the data is in the form of expressions. If the Expression check box is selected and the data begins with an equals sign (=), the software evaluates the data as an expression and returns the result. Otherwise, it returns the column value.</p>
<default_value_list>	<p>A comma-separated list containing the default expressions for the output columns. When no rows match the lookup condition, the default values are returned for the output column.</p> <p>Each default expression type must be compatible with the corresponding output column type such that if the types are not exactly the same, automatic conversion is still possible.</p> <p>If <default_value_list > is empty or has fewer expressions than the number of output columns, NULL is the default. You cannot have more default expressions than the number of output columns.</p>
<condition_list>	<p>A list of triplets that specify lookup conditions. Each set in a triplet contains <compare_column>, <compare_operator> (<, <=, >, >=, =, !=, IS, IS NOT, ~), and <compare_expression>.</p> <p>The <compare_column> is from the <lookup_table>. It is compared against <compare_expression> to compute the output row.</p> <p>The <compare_expression> is written in terms of constants, variables, and columns in the calling data flow or scripts. While it cannot contain column reference from the <lookup_table>, it can be a simple constant, variable, or column reference or a complex expression involving arithmetic operations and function calls.</p> <p>Use compare operators IS and IS NOT to examine <compare_column> against the NULL constant. When you use IS or IS NOT as the compare operator, <compare_expression> must contain the NULL constant. When you use other compare operators against a <compare_expression> containing a NULL, the lookup condition return value will always return FALSE. Use the compare operator ~ to indicate that the column from the lookup table contains a pattern. The required pattern tags in the lookup table are:</p> <ul style="list-style-type: none"> • mp (<pattern>) —Indicates the match_pattern type of pattern syntax • mr (<pattern>) —Indicates the match_regex type of pattern syntax • ms (<pattern>) —Indicates the match_simple type of pattern syntax <p>If <compare_expression> is an empty string, the function looks up a zero-length varchar value in the lookup table. The function ignores trailing blanks in the <compare_expression>.</p> <p>If you create more than one set of triplets, all triplets are implicitly joined with the AND operator to compute the final lookup condition.</p> <p>Example:</p> <pre>[c1, '=', 10, c2, '<', query.a, c3, '>=', lower(query.name)]</pre>

<code><orderby_column_list></code>	<p>A comma-separated list of column names from the <code><lookup_table></code>. Working together with <code><return_policy></code>, the <code><orderby_column_list></code> is used to determine which row to return the output when more than one row satisfies the lookup condition. When multiple rows occur, the list of rows is sorted based on columns from the <code><orderby_column_list></code> and choosing a row to return using the MIN/MAX <code><return_policy></code>.</p> <p>The <code><orderby_column_list></code> is optional. If you leave it blank, the orderby columns match the output columns.</p> <p>Examples:</p> <p><code>[c1,c3,c4]</code> — Sorts the rows using column values in c1, c3, c4.</p> <p><code>[]</code> — Indicated an empty list, which is a placeholder for specifying subsequent parameters.</p>
<code><output_variable_list></code>	<p>A comma-separated list of output variables. When more than one output column is specified in the function call, the output variables are used to receive output returns. Variables and output columns are matched by position.</p> <p>This parameter is optional unless more than one output column appears in the <code><return_column_list></code>. In the case of more than one output column, output variables must be equal in number to output columns.</p> <p>To enable conversion, the variable data type must be compatible with the corresponding output column. You do not need to specify output variables if the function is called using the function wizard to map output columns in the query window.</p> <p>Example: <code>[\$a,\$b,\$c]</code></p>

<sql_override>

This parameter, available as a button called *Custom SQL* in the function wizard, must contain a valid, single-quoted SQL SELECT statement or a \$variable of type VARCHAR to populate the lookup cache when the cache specification is PRE_LOAD_CACHE. This parameter replaces the SQL SELECT statement generated internally by the function for populating the cache. The SELECT statement must select at least those columns contained in <return_column_list>, <condition_list>, and <orderby_column_list>.

Any valid SQL SELECT statement is permitted and may contain references to other tables besides the <lookup_table> to specify inner and outer joins. This parameter can only be specified when the <lookup_table> is a database table.

If you specify this parameter with the cache specification of NO_CACHE, the software executes the sql_override query each time the function is called.

If you specify this parameter with the cache specification of PRE_LOAD_CACHE, only the first sql_override query is executed to populate the lookup cache. All subsequent SQL statements are ignored after the lookup cache is built.

If this parameter is specified when the cache specification is DEMAND_LOAD_CACHE, the caching mode will be converted to PRE_LOAD_CACHE and behave as if the PRE_LOAD_CACHE mode had been specified.

EXAMPLE:

```
[ 'select out1,
out2,compare1,compare2,orderby1,orderby2 from
lookuptbl,othertbl where c1=10
and
lookuptbl.c2=othertbl.c2' ]
```

Also, when you use <sql_override> in NO_CACHE mode, lookup_ext will dynamically execute the SQL if you pass in a dynamic SQL statement in the form of a variable.

The SET options include:

run_as_separate_process	Select (set to yes) to run each operation as a separate process (sub data flow) that uses separate resources (memory and computer) to improve performance and throughput. The default is no.
-------------------------	--

```
SET ("run_as_separate_process"='yes')
```

output_cols_info	Identifies whether an output column contains an expression (when the <i>Expression</i> check box is selected). The default is no. The syntax is as follows:
------------------	---

```
SET ("output_cols_info"='<?xml version="1.0"
encoding="UTF-8"?>
<output_cols_info>
<col index="1" expression="yes"/>
</output_cols_info>')
```

The first column begins with index value of 1.

Related Information

[match_pattern \[page 1164\]](#)

[match_regex \[page 1166\]](#)

[match_simple \[page 1171\]](#)

9.3.94.1 lookup_ext optimization and limitations

You can optimize lookup_ext on databases, and there are some limitations to using this function.

Optimizing database push-down

For best performance, the lookup_ext function can be pushed down to the database when the following conditions are met:

- The lookup_ext function is used in the column mapping, output schema, or `SELECT WHERE` clause of a Query transform.
- The `<lookup_table>` is a database table from the same datastore or a linked datastore as the reader.
- The `<cache_spec>` is set to `NO_CACHE`.
- The `<return_policy>` is set to either `MAX` or `MIN`.
- All conditions used in the `<condition_list>` are database expressions.
- Only the equals operator (`=`) are used in the lookup `<condition_list>`.
- The `<run_as_separate_process>` SET option is set to `no`.
- For lookups with multiple-result column values, the database must support the rank (or equivalent) function.

Note

For SAP HANA, MySQL, SAP ASE, and Informix databases, no analytic function support is available. As a result, push-down is supported in all cases for single-result columns, and multiple-result columns only for primary keys.

Limitations

- You cannot call the lookup_ext function from an ABAP data flow; use the lookup() function as an alternative.
- When calling lookup_ext in script objects, jobs, or work flows, the caching mode is always `NO_CACHE` because the software cannot determine when to release the cache after executing the function.
- In all parameters of lookup_ext, you can refer to any supported data type except the long, blob, and NRDM data types. Therefore, you cannot look up a long column or specify a column or expression of the long data type in `<default_value_list>`, `<orderby_column_list>`, or `<condition_list>`.

- If an optional parameter is missing, an empty placeholder ([]) must occupy that position if other optional parameters that follow the missing parameter are specified in the function call.
- It's recommended that for best performance you use the equals operator to specify the lookup condition. If the caching mode is NO_CACHE and the lookup_ext is against a database table, the underlying DBMS typically has fast access methods such as an index to retrieve data based on an indexed key. When the caching method is PRE_LOAD_CACHE, using the equals comparison will result in more efficient memory lookup than any other comparison operators.
If the caching method is PRE_LOAD_CACHE, any lookup condition involving a constant expression will be pushed down to the database, resulting in a smaller lookup cache than the current lookup.
- Pattern evaluation uses virtual memory and is not included as part of pageable cache when the lookup table is cached. So if the lookup table has a lot of patterns, then the data flow could run out of memory. In those cases, select the [Run as a separate process](#) check box. This limitation also applies when using expressions in output columns.

9.3.95 lookup_seq

Retrieves a value in a table or file based on the values in a different source table or file and a particular sequence value.

≡ Syntax

```
lookup_seq (<lookup_table>, <result_column>, <default_value>,  
<sequence_column>, <sequence_expression>, <compare_column>, <expression>)
```

Return value

any type

The value in the <lookup_table> that meets the lookup_seq requirements. The return type is the same as <result_column>.

Where

<code><lookup_table></code>	<p>The table or file that contains the result or value you are looking up (<code><result_column></code>). The <code><sequence_column></code> and <code><compare_column></code> are also located in this table.</p> <p>Use a fully qualified table name that includes the datastore, owner, and table name. For example: ERP_ds.OWNER.EMPLOYEES</p> <p>For Netezza 7.x multi-schema, include the datastore, owner, schema and table name. For example: NZ7_ds.DSDEV.SCHEMA1.TABLE1. If the schema name SCHEMA1 is not specified, the default schema is used. For example: NZ7_ds.DSDEV.TABLE1.</p> <p>The <code><lookup_table></code> is cached automatically for the operation of the function.</p>
<code><result_column></code>	<p>The column containing the values you want to retrieve. This column is in the <code><lookup_table></code>.</p> <p>When the column contains varchar values, the function does not trim trailing blanks.</p>
<code><default_value></code>	<p>The value returned when there is no matching row in the <code><lookup_table></code>.</p>
<code><sequence_column></code>	<p>The column in <code><lookup_table></code> that indicates the sequence of the row. This column often contains a date that indicates when new values were added to the row. For example, in some source tables, <code><sequence_column></code> is the EFFDT column, which indicates when the data in the row became effective.</p>
<code><sequence_> <expression></code>	<p>The value the function searches for in the <code><sequence_column></code> to find a matching row. For example, if you are looking up values from a slowly changing dimension table and are interested in only those rows in which the data is current as of today, you could use the return value from the sysdate function for <code><sequence_expression></code>.</p> <p>If <code><sequence_expression></code> is an empty string, the function looks up a zero-length varchar value in the lookup table.</p>
<code><compare_column></code>	<p>The column in the <code><lookup_table></code> that the function uses to find a matching row.</p>
<code><expression></code>	<p>The value that the function searches for in the <code><compare_column></code>. This can be a simple column reference, such as a column found in both a source and the <code><lookup_table></code>. This can also be a complex expression given in terms of constants and input column references.</p> <p>When <code><expression></code> refers to a unique source column, you do not need to include a table name qualifier. If <code><expression></code> is from another table or is not unique among the source columns, you need a table name qualifier.</p> <p>If <code><expression></code> is an empty string, the function looks up a zero-length varchar value in the lookup table.</p> <p>The function ignores trailing blanks in comparisons of <code><expression ></code> and values in <code><compare_column></code>.</p>

i Note

You can specify more than one `<compare_column>` and `<expression>` pair—simply add additional pairs at the end of the function statement. The values must match for all specified pairs in order for the lookup function to find a matching row.

The `lookup_seq` function uses a value you provide (`<expression>`) to find a corresponding value in a different file or table (`<lookup_table>`). When multiple rows match, the function uses the row's sequence to determine the matching row.

More specifically, the function searches for the rows in the `<lookup_table>` where the value in the `<compare_column>` matches the value in `<expression>`. When the function finds multiple matching rows, it searches in the `<sequence_column>` for the row with the closest value less than or equal to the `<sequence_expression>`. If no row has a value less than or equal to the `<sequence_expression>`, the function finds the row with the closest value to the `<sequence_expression>`. For the matching row, the function returns the value in the `<result_column>`.

For example, if your source schema uses an employee ID to identify each row, and you want the employee's salary at the end of the previous year in your target schema, you can use the `lookup_seq` function to return the employee salary given the employee ID and the effective date of the salary. The salary returned will be the value corresponding to the latest effective date less than or equal to the value of `<sequence_expression>`.

In SQL terms, the `lookup_seq` function evaluates `<expression>` for each row, then determines which sequence column value meets the requirements:

```
SELECT MAX(<sequence_column>)
  FROM <lookup_table>, <source_table>
 WHERE <sequence_column> <= <sequence_expression>
    AND <compare_column> =
      <source_table.expression>
```

Suppose this query stores the `<sequence_column>` value returned as `<sequence_result>`. Next, the function uses the `<sequence_result>` to find the proper `<result_column>`:

```
SELECT <result_column>
  FROM <lookup_table>, <source_table>
 WHERE <sequence_column> = <sequence_result>
    AND <compare_column> =
      <source_table.expression>
```

The value returned by these queries is the result of the `lookup_seq` function for the row.

You can specify multiple `<compare_column>` and `<expression>` pairs to uniquely identify the `<result_column>` value. However, the function wizard only provides fields for one pair; add extra `<compare_column>` and `<expression>` pairs to the output that the wizard generates.

Data Services always caches the comparison table when performing a `lookup_seq` function.

If the `lookup_seq` function does not find the value of `<expression>` in `<compare_column>`, then the function evaluates and returns the `<default_value>`.

❖ Example

You can use the `lookup_seq` function to return a value from a slowly changing dimension table given an identifier. For example, suppose you have a source table that contains a numerical identifier, such as an employee number, and you want to retrieve the employee's salary at a specific time in the past. You can use

the `lookup_seq` function to return the employee's salary on a particular date based on the employee number.

The source table contains the employee number and employee name.

You want the target table to contain the employee name and salary.

Use the `lookup_seq` function to translate the values from the source table to the desired values in the target table. The `lookup_seq` function uses a third, "translation," table.

To produce the desired target column, select the column in the target schema. Next, click the [Functions](#) button, located over the [Mapping](#) text box. In the function wizard, select [Miscellaneous_functions](#) under Function categories, then select [lookup_seq](#) under Function name. Enter the function parameters as follows:

Option	Value
Translate table	ODS_DS.SSB.FINANCE
Result column	Salary
Default value	'0'
Sequence column	EffectiveDate
Sequence Expression	'12.31.1999'
Compare column	EmployeeID
Expression	EmployID

The function wizard automatically produces the mapping text

```
lookup_seq(Ora_DS.RBH.FINANCE, SALARY, '0',  
           EFFECTIVEDATE, '12.31.1999', Employee_ID, EmployID)
```

For each employee, this `lookup_seq` function returns the value from the salary column for that employee that is the most recent before December 31, 1999.

9.3.96 lower

Changes the characters in a string to lowercase.

Syntax

```
lower(<value>, '<locale>')
```

Return value


varchar

The lowercase string. The return type is the same as `<value>`. Any characters that are not letters are left unchanged.

Where

<code><value></code>	The string to be modified.
<code><locale></code>	Optional parameter that converts the string to the specified locale.

i Note
ISO 639 language code and ISO 3166 country code formats are supported.

 **Example**

Function	Results
<code>lower ('Accounting101')</code>	'accounting101'
<code>upper ((LastName,1,1)) lower (substr (LastName, 2, LENGTH (LastName)))</code>	The value in column <code>LastName</code> with the first letter uppercase and the rest of the value lowercase. Note that this example does not account for two-word last names.
<code>lower (LastName, 'tr')</code>	The value in column <code>LastName</code> is converted to all lowercase. It is also converted to the Turkish locale, using the ISO 639 language code.


Related Information

- [ISO 639 language list](#)
- [ISO 3166 Country Code list](#)

9.3.97 lpad

Pads the string with characters in the left from a given pattern.

This function repeats the pattern at the beginning of the input string until the final string is the appropriate length. If the input_string is already longer than the expected length, then this function truncates the string.

 **Syntax**

`lpad (<input_string>, <size>, '<pad_string>')`

Return value

varchar

The modified string. The return type is the same as `<value>`. Any characters that are not letters are left unchanged.

Where

<code><input_string></code>	The string source.
<code><size></code>	An integer value indicating the number of characters in the return string.
<code><pad_string></code>	A character or set of characters that this function concatenate to <code><input_string></code> .

❖ Example

Function	Results
<code>lpad('Tanaka', 15, '')</code>	<code>' Tanaka'</code>
<code>lpad(last_name, 25, '')</code>	The value in the column last_name, padded with spaces to 25 characters on the left or truncated to 25 characters.

9.3.98 lpad_ext

Pads the left side of the string with logical characters from a given pattern.

i Note

These logical characters prohibit this function from getting pushed down to the database.

This function repeats the pattern at the beginning of the input string until the final string is the appropriate length. If the input_string is already longer than the expected length, then this function truncates the string.

≡ Syntax

```
lpad_ext(<input_string>, <size>, '<pad_string>')
```

Return value

varchar

The modified string. The return type is the same as `<value>`. Any characters that are not letters are left unchanged.

Where

<code><input_string></code>	The string source.
<code><size></code>	An integer value indicating the number of characters in the return string.
<code><pad_string></code>	A logical character or set of logical characters that this function concatenates to the <code><input_string></code> .

❖ Example

Function	Results
<code>lpad_ext('Tanaka', 15, '')</code>	' Tanaka '
<code>lpad_ext(last_name, 25, '')</code>	The value in the column last_name, padded with spaces to 25 characters on the left or truncated to 25 characters.

`lpad_ext` and `lpad` functions exhibit the same behavior when the functions are evaluated. In situations where the function is pushed down to the database, the database behavior may differ when `'input_string'` and/or `'pad_string'` parameters contain multibyte characters.

Function	Input	Output
<code>lpad</code>	<code>("abc𐀀𐀀", 10, '')</code>	' abc𐀀𐀀 '
<code>lpad_ext</code>	<code>("abc𐀀𐀀", 10, '')</code>	' abc𐀀𐀀 '
<code>lpad</code>	<code>("abc𐀀𐀀", 10, "")</code>	' abc𐀀𐀀 '
<code>lpad_ext</code>	<code>("abc𐀀𐀀", 10, "")</code>	' abc𐀀𐀀 '
<code>lpad</code>	<code>("abcd", 10, '𐀀')</code>	' 𐀀𐀀𐀀abcd '
<code>lpad_ext</code>	<code>("abcd", 10, '𐀀')</code>	' 𐀀𐀀𐀀𐀀abcd '

9.3.99 ltrim

Removes specified characters from the start of a string.

Syntax

```
ltrim(<input_string>, <trim_string>)
```

Return value

varchar

The modified string. The return type is the same as `<input_string>`.

Where

<code><input_string></code>	The string to be modified.
<code><trim_string></code>	The characters to remove from <code><input_string></code> .

The ltrim function is case-sensitive.

The function scans `<input_string>` left-to-right removing all characters that appear in `<trim_string>` until it reaches a character not in `<trim_string>`.

Example

Function	Results
<code>ltrim('Marilyn', ' ')</code>	<code>'Marilyn'</code>
<code>ltrim('ABCABCD', 'ABC')</code>	<code>'D'</code>
<code>ltrim('ABCABCD', 'EFG')</code>	<code>'ABCABCD'</code>
<code>ltrim('ABCDEABCDE', 'ABC')</code>	<code>'DEABCDE'</code>

To remove all leading blanks in a string, use ltrim as follows:

```
ltrim(EMPLOYEE.NAME, ' ')
```

where `EMPLOYEE.NAME` specifies the NAME column in the EMPLOYEE table.

9.3.100 ltrim_blanks

Removes blank characters from the start of a string.

Syntax

```
ltrim_blanks(<input_string>)
```

Return value

varchar

The modified string. The return type is the same as `<input_string>`.

Where

`<input_string>`

The string to be modified.

Example

Function	Results
<code>ltrim_blanks(' Marilyn')</code>	<code>'Marilyn'</code>
<code>ltrim_blanks(last_name)</code>	The value contained in the column last_name, with all leading blanks and control characters removed.

9.3.101 ltrim_blanks_ext

Removes blank and control characters from the start of a string.

Syntax

```
ltrim_blanks_ext(<input_string>)
```

Return value

varchar

The modified string. The return type is the same as `<input_string>`.

Where

`<input_string>`

The string to be modified.

❖ Example

Function	Results
<code>ltrim_blanks_ext(' Marilyn')</code>	<code>'Marilyn'</code>
<code>ltrim_blanks_ext(last_name)</code>	The value contained in the column last_name, with all leading blanks removed.

9.3.102 mail_to

Captures the specified number of lines in the trace log and error log, packages the information as e-mail, and uses your Job Server computer's mail client to send e-mail messages to your local mail server for standard e-mail processing.

≡ Syntax

```
mail_to(<recipients_list>, <subject>, <message>, <number_of_trace_lines>,  
        <number_of_error_lines>)
```

Return value

int

Returns 0 if function succeeds. Returns a non-zero integer if function fails.

Where

<code><recipients_list></code>	A string containing one or more recipient e-mail addresses separated by commas (.). This string cannot be empty and must contain valid, qualified e-mail address information.
<code><subject></code>	A string containing the subject of the e-mail. This string can be empty.
<code><message></code>	A string containing the e-mail message. This string can be empty.
<code><number_of_trace> <_lines></code>	The number of lines from the end of the trace log file to append to the end of the e-mail. This input cannot be empty.
<code><number_of_error> <_lines></code>	The number of lines from the end of the error log file to append at the end of the e-mail. This input cannot be empty.

Only use this function within a script.

To use this function, a mail client must be installed and running on the Job Server computer that calls the function. The login account for the mail client must have the same user name and password as the SAP Data Services service. The type of client varies by the operating system:

- If the Job Server is on a computer running the Windows operating system, then the mail client must comply with MAPI (message application programming interface). In addition, the mail client must be configured as the default mail client. For example, Microsoft Outlook is a MAPI-based mail client.
- If the Job Server is on a computer running the UNIX operating system, then the mail client must be mailx-compliant.

Note

If you do not have the required mail client, contact SAP Business User Support for assistance.

❖ Example

Function	Results
<pre>\$myvar = mail_to('admin@company.com', 'Out of memory error in the SalesFact job. Please fix the error before running recovery job.', ' ', 10, 10)</pre>	The message is sent to one recipient.

Function	Results
<pre>\$myvar = mail_to('admin@company.com, manager@company.com', 'Out of memory error:' systime(), 'Out of memory error while running the data flow:' \$dataflow_name ' in the job:' \$job_name '.', 10, 20)</pre>	<p>The message is sent to two recipients. The job name and data flow names are included in the text of the message as variables. Note that the software trims blank spaces from the end of strings; this example includes a blank on the beginning of the next string. You can also concatenate a string with a single blank.</p> <p>In the script, type "\$a = ;" where \$a is the local integer variable defined in the work flow. Put the cursor just ahead of the semicolon before clicking the Functions button to construct a mail_to statement.</p>

i Note

Often, you list e-mail addresses as nicknames in your mail service address book. If your mail system is compatible with the software mail_to function, you can use these nicknames (comma separated) as values in the `<recipients_list>`. In this case, the software mailer program searches your e-mail address book for the nickname and uses the corresponding qualified e-mail address for message routing.

Limitation

The mail_to function might not work properly on Windows 2000 after downloading a security patch for Microsoft Outlook that includes email prompts. In this case, you have two options:

- To change the Outlook security settings to suppress prompts, see the instructions for Microsoft article 263297 on the following Web site: <http://support.microsoft.com/default.aspx?scid=kb;en-us;Q263297&id=263297&SD=MSKB#OB>.
The article describes how to suppress prompting when sending e-mail from a particular computer. In summary, first create a security policy file on the Exchange Server (usually done by an administrator). In the security policy file, turn off prompting. Then add a registry key to the Job Server (client) computer. When the Job Server computer tries to send an e-mail, the Outlook client first checks the registry key; if the key is set, Outlook checks the security policy file on the Exchange Server and suppresses prompts.
- Use the smtp_to function instead.

Related Information

[Microsoft Help and Support](#) ➔

9.3.103 match_pattern

Allows you to match a whole input string to simple patterns that Data Services supports for this function. This function does not match substrings.

≡ Syntax

```
match_pattern(<input_string>,<pattern_string>)
```

Return Value

integer	Returns 1 for a match, otherwise 0.
---------	-------------------------------------

Where

input_string	String to be matched. Supports UNICODE characters.
pattern_string	Pattern you want to find in a whole input string. Substring matches are not supported.

Use the following characters to create a pattern:

X	Represents uppercase characters; general category Lu as per Unicode 4.0 specification (for example, Latin, Greek, Cyrillic, Armenian, Deseret, and archaic Georgian).
x	Represents non-uppercase characters: <ul style="list-style-type: none">• Ll—Lowercase letter (for example Latin, Greek, Cyrillic, Armenian, Deseret, and archaic Georgian).• Lt—Title letters (for example Latin capital letter D with small letter Z).• Lm— Modifier letter (for example acute accent, grave accent).• Lo—Other letters (including Chinese, Japanese, and so on).
9	Represents numbers.
\	Escape character.
*	Any characters occurring zero or more times.
?	Any single character occurring once and only once.
[]	Any one character inside the braces occurring once.

[!] Any character except those after the exclamation point (i.e. [!12] can allow any, say zip code, that does not start with a 1 or 2.

All other characters represent themselves. If you want to specify a special character as itself, then it has to be escaped. For example, [!9] means except any digit. To specify except nine, the correct pattern is [!\9].

The following table displays pattern strings that represent example values:

Example Value	Pattern string
Henrick	Xxxxxxx
DAVID	XXXXX
Tom Le	Xxx Xx
Real-time	Xxxx-xxxx
JJD)\$@&*hhN8922hJ7#	XXX)\$@&*xxX9999xX9#
1,553	9,999
0.32	9.99
-43.88	-99.99
Returns names with last name Jones	*Jones
Returns Henrick1 or HenrickZ	Henrick?
Returns David1 or David2 or David3	David[123]

❖ Example

Use the `match_pattern` function in the Validation transform or in a WHERE clause of Query. The input string can be from sources such as columns, variables, or constant strings.

Use Case	Pattern	Function Call	Results
To match a zip code except one that begins with 1 or 2.	'[!12]9999'	<pre>if (match_pattern(' 15014', '[! 12]9999') <> 0) print('matched') ; else print('not matched');</pre>	Function prints "not matched".

Use Case	Pattern	Function Call	Results
To match a zip code except one that begins with 1 or 2.	'[!12]9999'	<pre> if (match_pattern(' 55014', '[! 12]9999') <> 0) print('matched') ; else print('not matched'); </pre>	Function prints "matched".
To process only customer phone numbers that fit the same pattern.	'999-999-9999'	<pre> WHERE MATCH_PATTERN(CU STOMER.PHONE_NUM ,'999-999-9999') <> 0 </pre>	Phone numbers that do not match the pattern throw error 0.
To check a string against a complex pattern and print result to trace log.	'XXX)\$@&*xxX9999xX9#'	<pre> if (match_patter n('JJD) \$@&*hhN8922hJ7# ', 'XXX) \$@&*xxX9999xX9# ') <> 0) print ('matched'); else print('not matched'); </pre>	The result for this call is "matched".

Related Information

[literal](#) [page 1132]

9.3.104 match_regex

Matches whole input strings to the pattern that you specify with regular expressions (regular expressions based on the POSIX standard) and flags. POSIX refers to the POSIX.1 standard (IEEE Std 1003.1) which defines system interfaces and headers with relevance for string handling and internationalization. The XPG3, XPG4, Single Unix Specification (SUS) and other standards include POSIX.1 as a subset. The patterns listed here in the *Reference Guide* adhere to the current standard. See <http://icu.sourceforge.net/userguide/regexp.html> for more information and updates. This function does not match substrings.

≡ Syntax

```
match_regex (<input_string>, <regular_expression_pattern>, <flags>)
```


Return Value

integer	Returns 1 for a match, otherwise 0.
---------	-------------------------------------

Where

<code><input_string></code>	String to be matched. Supports UNICODE characters.
<code><regular_expression _pattern></code>	Pattern you want to find in a whole input string. Substring matches are not supported. Provide the pattern in regular expression format with a varchar data type.
<code><flags></code>	Allows you to specify additional behavior that you want to occur while Data Services searches the input_string for pattern matches. Enter NULL if you do not want to specify a flag. You can combine the options for flags using a comma. Flag options are case sensitive and need to be specified in upper case.

You can use the following regular expression patterns in the pattern parameter:

Character	Description
<code>\a</code>	Match a BELL, \u0007.
<code>\A</code>	Match at the beginning of the input. Differs from <code>^</code> in that <code>\A</code> will not match after a new line within the input.
<code>\b</code> , outside of a <code>[Set]</code>	Match if the current position is a word boundary. Boundaries occur at the transitions between word (<code>\w</code>) and non-word (<code>\W</code>) characters, with combining marks ignored. For better word boundaries, see ICU Boundary Analysis.
<code>\b</code> , within a <code>[Set]</code>	Match a BACKSPACE, \u0008.
<code>\B</code>	Match if the current position is not a word boundary.
<code>\cX</code>	Match a control-X character.
<code>\d</code>	Match any character with the Unicode General Category of Nd (Number, Decimal Digit).
<code>\D</code>	Match any character that is not a decimal digit.
<code>\e</code>	Match an ESCAPE, \u001B.
<code>\E</code>	Terminates a <code>\Q ... \E</code> quoted sequence.
<code>\f</code>	Match a FORM FEED, \u000C.

Character	Description
\G	Match if the current position is at the end of the previous match.
\n	Match a LINE FEED, \u000A.
\N{UNICODE CHARACTER NAME}	Match the named character.
\p{UNICODE PROPERTY NAME}	Match any character with the specified Unicode Property.
\P{UNICODE PROPERTY NAME}	Match any character not having the specified Unicode Property.
\Q	Quotes all following characters until \E.
\r	Match a CARRIAGE RETURN, \u000D.
\s	Match a white space character. White space is defined as [\t\n\f\r\p{Z}].
\S	Match a non-white space character.
\t	Match a HORIZONTAL TABULATION, \u0009.
\uhhhh	Match the character with the hex value hhhh.
\Uhhhhhhh	Match the character with the hex value hhhhhhhh. Exactly eight hex digits must be provided, even though the largest Unicode code point is \U0010ffff.
\w	Match a word character. Word characters are [\p{Li}\p{Lu}\p{Lt}\p{Lo}\p{Nd}].
\W	Match a non-word character.
\x{hhhh}	Match the character with hex value hhhh. From one to six hex digits may be supplied.
\xhh	Match the character with two digit hex value hh.
\X	Match a Grapheme Cluster.
\Z	Match if the current position is at the end of input, but before the final line terminator, if one exists.
\z	Match if the current position is at the end of input.
\n	Back Reference. Match whatever the nth capturing group matched. n must be a number > 1 and < total number of capture groups in the pattern. Note: Octal escapes, such as \012, are not supported in ICU regular expressions.
[pattern]	Match any one character from the set. See UnicodeSet for a full description of what may appear in the pattern.

Character	Description
.	Match any character.
^	Match at the beginning of a line.
\$	Match at the end of a line.
\	Quotes the following character. Characters that must be quoted to be treated as literals are * ? + [() { } ^ \$ \ . /

You can use the following regular expression operators in a pattern parameter:

Operator	Description
	Alternation. A B matches either A or B.
*	Match 0 or more times. Match as many times as possible.
+	Match 1 or more times. Match as many times as possible.
?	Match zero or one times. Prefer one.
{n}	Match exactly n times.
{n,}	Match at least n times. Match as many times as possible.
{n,m}	Match between n and m times. Match as many times as possible, but not more than m.
*?	Match 0 or more times. Match as few times as possible.
+	Match 1 or more times. Match as few times as possible.
??	Match zero or one times. Prefer zero.
{n}?	Match exactly n times.
{n,}?	Match at least n times, but no more than required for an overall pattern match.
{n,m}?	Match between n and m times. Match as few times as possible, but not less than n.
*+	Match 0 or more times. Match as many times as possible when first encountered, do not retry with fewer even if overall match fails. Possessive match.
++	Match 1 or more times. Possessive match.
?+	Match zero or one times. Possessive match.
{n}+	Match exactly n times.
{n,}+	Match at least n times. Possessive match.

Operator	Description
<code>{n,m}+</code>	Match between n and m times. Possessive match.
<code>(...)</code>	Capturing parentheses. Range of input that matched the parenthesized subexpression is available after the match.
<code>(?: ...)</code>	Non-capturing parentheses. Groups the included pattern, but does not provide capturing of matching text. Somewhat more efficient than capturing parentheses.
<code>(?> ...)</code>	Atomic-match parentheses. First match of the parenthesized subexpression is the only one tried; if it does not lead to an overall pattern match, back up the search for a match to a position before the "(?>"
<code>(?# ...)</code>	Free-format comment (<code>?# comment</code>).
<code>(?= ...)</code>	Look-ahead assertion. True if the parenthesized pattern matches at the current input position, but does not advance the input position.
<code>(?! ...)</code>	Negative look-ahead assertion. True if the parenthesized pattern does not match at the current input position. Does not advance the input position.
<code>(?<= ...)</code>	Look-behind assertion. True if the parenthesized pattern matches text preceding the current input position, with the last character of the match being the input character just before the current position. Does not alter the input position. The length of possible strings matched by the look-behind pattern must not be unbounded (no * or + operators).
<code>(?<! ...)</code>	Negative look-behind assertion. True if the parenthesized pattern does not match text preceding the current input position, with the last character of the match being the input character just before the current position. Does not alter the input position. The length of possible strings matched by the look-behind pattern must not be unbounded (no * or + operators).
<code>(?ismx-ismx: ...)</code>	Flag settings. Evaluate the parenthesized expression with the specified flags enabled or disabled.
<code>(?ismx-ismx)</code>	Flag settings. Change the flag settings. Changes apply to the portion of the pattern following the setting. For example, <code>(?i)</code> changes to a case insensitive match.

You can use the following flags in the flag parameter:

Flag Options	Description
'CASE_INSENSITIVE'	If set, matching will take place in a case-insensitive manner.
'COMMENTS'	If set, allows use of white space and #comments within patterns.
'DOTALL'	If set, a "." in a pattern will match a line terminator in the input text. By default, it will not. Note that a carriage-return / line-feed pair in text behave as a single line terminator and match a single "." in a regular expression pattern.
'MULTILINE'	If set, the input string is treated as multiple lines instead of a single line, and the '^' and '\$' characters apply to each line in the input string instead of the entire input string.

Flag Options	Description
'NO_PUSHDOWN'	If set, the match_regex function will be processed but will never be pushed down to the data-base.

❖ Example

Use the Match_regex function in the Validation transform by accessing the Smart Editor or Function wizard or in a WHERE clause of a Query. The input string can be from sources such as columns, variables, or constant strings.

Use Case	Pattern	Function Call
To match phone numbers in (408)-933-6000 format.	'([0-9]{3}-[0-9]{3}-[0-9]{4})'	<code>match_regex (pho_number, '([0-9]{3}-[0-9]{3}-[0-9]{4})', NULL)</code>
To match a string that starts with "topicA" regardless of case.	'topicA.*'	<code>match_regex (subject, 'topicA.*', 'CASE_INSENSITIVE')</code>

Related Information

[literal \[page 1132\]](#)

9.3.105 match_simple

Allows you to match a whole input string to simple patterns that Data Services supports for this function. This function does not match substrings.

≡ Syntax

```
match_simple(<input_string>,<pattern_string>)
```

Return Value

integer	Returns 1 for a match, otherwise 0.
---------	-------------------------------------

Where

<code><input_string></code>	String to be matched. Supports UNICODE characters.
<code><pattern_string></code>	Pattern you want to find in a whole input string.

Use the following characters to create a pattern:

.	Represents any single character.
*	Represents any character zero or more times.
#	Represents any single alphabetic character including non-English letters.
\$	Represents any alphabetic character, including non-English letters, zero or more times.
+	Matches the previous character one or more times.
(string)+	Matches the string one or more times.
[number1..number2]	Numeric range (integers only). Matches any number between number1 and number2.
\	Escape character
;	OR operator. If the data matches any of the identified patterns, the result is TRUE. Enclose the list with curly brackets {}. Example: <code>{ABC+;XYZ*}</code> If the data matches either ABC+ or XYZ*, the result is TRUE.
<>	NOT operator. Specify the pattern after the <>. Example: <code><><pattern></code>
{EMPTY} and {empty}	Special predefined patterns that match empty data.
{NULL} and {null}	Special predefined patterns that match NULL data.

If the pattern is empty, then it matches all data.

If the value of a pattern column is NULL, then it will not match with any value.

All other characters represent themselves. If you want to specify a special character as itself, then it has to be escaped.

The following table displays pattern strings that represent example values:

Example Value	Pattern string
ACCT1234567	ACCT*
ZIP10000 to ZIP99999	ZIP[10000..99999]
ACCT123 or ACCOUNT234	{ACCT*;ACCOUNT*}
www.anything.com	www\$.com

❖ Example

Use the `match_simple` function in the Validation transform or in a WHERE clause of Query. The input string can be from sources such as columns, variables, or constant strings. The following example illustrates sample code used in a script.

Use Case	Pattern	Function Call	Results
To match account numbers from ACCT1 to ACCT5000	ACCT[1..5000]	<pre>if (match_simple('ACCT14', 'ACCT[1..5000]')) <> 0 print ('matched'); else print('not matched');</pre>	Function prints "matched".

9.3.106 max

Returns the maximum value from a list.

≡ Syntax

```
max(<value_list>)
```

Return value

any type

The maximum value of the column values. The return type is the same as the values in `<value_list>`.

Where

`<value_list>`

The source values for which to identify a maximum.

❖ Example

To calculate the maximum value in the salary column of a table, use the `max` function in a query:

- In the [Mapping](#) tab of the query editor, enter:

```
max (SALARY)
```

- In the [Group By](#) tab in the query editor, specify the columns for which you want to find the maximum salary, such as the department column. For each unique set of values in the group by list, such as each unique department, Data Services calculates the maximum salary.

9.3.107 min

Returns the minimum value from a list.

Syntax

```
min(<value_list>)
```

Return value

any type

The minimum value of the column values. The return type is the same as the values in `<value_list>`.

Where

`<value_list>`

The source values for which to identify a minimum.

Example

To calculate the minimum value in the salary column of a table, use the min function in a query:

- In the *Mapping* tab of the query editor, enter:

```
min(SALARY)
```

- In the *Group By* tab in the query editor, specify the columns for which you want to find the minimum salary, such as the department column. For each unique set of values in the group by list, such as each unique department, Data Services calculates the minimum salary.

9.3.108 mod

Returns the remainder when one number is divided by another.

Syntax

```
mod(<number1>, <number2>)
```


Return Value

Depends on the input numbers.

Where

<code><number1></code>	Number to be divided.
<code><number2></code>	Divisor of first number.

Details

Returns the remainder when one number is divided by another.

Note that the % operator used in syntax produces the same result.

❖ Example

Function	Result
<code>mod(100.10,10) ;</code>	0.100000

9.3.109 month

Determines the month in which the given date falls.

≡ Syntax

```
month(<date1>)
```

Return value

int

The number from 1 to 12 that represents the month component of `<date1>`.

Where

<date1>

The source date.

❖ Example

Function	Results
<code>month(to_date('Jan 22, 1997', 'mon dd, yyyy'))</code>	1
<code>month(to_date('3/97', 'mm/yy'))</code>	3

9.3.110 num_to_interval

Converts a numeric value to an interval.

≡ Syntax

```
num_to_interval(<number1, format>)
```

Return value

interval

The converted interval.

Where

<number1>

The value of type `int`, `real`, `decimal`, or `numeric` to convert.

<format>

A string describing the format of the interval. Choose from the following values:

D: Days

H: Hours

M: Minutes

S: Seconds

❖ Example

Function	Results
<code>num_to_interval(elapsed_days, 'D')</code>	The value from the column <code>elapsed_days</code> converted to an interval of days.
<code>start_time + num_to_interval(elapsed_seconds, 'S')</code>	This example assumes that it is acting on an input schema which contains (at least) the columns 'start_time' and 'elapsed_seconds' (for example, the start_time might be '2005-12-01 00:00:00' and elapsed_seconds might be 200). So, this example indicates a time which is the number of elapsed seconds after the start time ('2005-12-01 00:03:20').

9.3.111 nvl

Replaces NULL values.

≡ Syntax

```
nvl(<expression1>, <replacement_value>)
```

Return value

any type

The value of <expression1> if not NULL, otherwise, the value of <replacement_value>.

Where

<expression1>	The value to be tested for NULL.
<replacement_value>	The value to replace <expression1> if <expression1> is NULL. <replacement_value> must be the same data type as <expression1>.

Example

Function	Results
<code>nvl(modification_date, sysdate())</code>	If the column <code>modification_date</code> for a row hasn't been set, this function inserts today's date.
<code>nvl(lookup(r3..vbpa, kunnr, 'NULL', vbeln, vbak.vbeln, posnr, vbap.posnr, parvw, 'RE'), lookup(r3..vbpa, kunnr, 'NULL', vbeln, vbak.vbeln, posnr, vbap.posnr, parvw, 'RG'))</code>	Both expressions are determined by the result of lookup functions.

9.3.112 power

Returns the value of the given expression to the specified power.

Syntax

```
power(<num>, <num>)
```

Return Value

Float data type

Where

<num>	Numeric expression representing base number.
-------	--

<num>

Numeric expression representing power.

Details

Returns the value of the given expression to the specified power.

Example

Function	Results
<code>power(2.2,3);</code>	10.648000

9.3.113 previous_row_value

Returns the column value of the previous row.

Note

Each call to the `previous_row_value()` function will return the value stored during the previous call of this function. In other words, if the function is not called for each row, the results of this function might not be what you expect (not the previous row's value). This scenario can happen if you, for example, use the `previous_row_value()` inside an `ifthenelse()` function:

```
If_then_else (table1.status = 'new', 0 , previous_row_value(table1.value))
```

A better solution in the example above would be :

```
If_then_else (table1.status = 'new', 0 , 1) * previous_row_value(table1.value)
```

or use two different queries: one for the `previous_row_value()` and one for the final result including the `if_then_else()`.

Syntax

```
previous_row_value(<expression>)
```

Return Value

Data type of the input parameter. First row always returns `NULL`.

Where

<expression>

Valid Input expression.

Details

This function is useful in Query transforms. It returns the previous row's value. For example, the input stream of the column might be 1; 2; 3; 4 for the first four rows. The function returns `NULL; 1; 2; 3`.

❖ Example

Following is list of records of sales figures for each day:

Date Revenue

rec 1 1/1/2005 1000

rec 2 1/2/2005 1100

rec 3 1/3/2005 900

rec 4 1/4/2005 1200

The requirement is to calculate the delta of the revenue with the previous day. So the query uses an order by on Date and calculate Revenue - Previous_Row_Value (revenue) which results in:

Date Revenue Delta = Revenue - Previous_Row_Value

rec 1 1/1/2005 1000 NULL

rec 2 1/2/2005 1100 +100

rec 3 1/3/2005 900 -200

rec 4 1/4/2005 1200 +300

9.3.114 print

Prints the given string to the trace log.

≡ Syntax

```
print('<input_string>')
```

Return value

int

Value is `<input_string>` when the string contains valid data. Value is NULL and no string prints when the string contains NULL data.

Where

`<input_string>`

The message to be written to the trace log.

❖ Example

Function	Results
<pre>print('Reached decision point for running full or incremental data flows')</pre>	Writes "Reached decision point for running full or incremental flows" to trace log and returns <code><input_string></code> .
<pre>print('The date is: [\$start_date]')</pre>	Writes "The date is 2000.06.03" to trace log and returns <code><input_string></code> .
<pre>print('[\$month_sal*12]')</pre>	Writes "48000" to trace log and returns <code><input_string></code> .
<pre>print('Total Sal is: [\$month_sal*12]');</pre>	Writes "Total Sal is: 48000" to trace log and returns <code><input_string></code> .
<pre>print('The return value from the SQL() function is > [\$y]');</pre>	Writes "The return value from the SQL() function is > 23456" to trace log and returns <code><input_string></code> .

9.3.115 pushdown_sql

Allows you to create dynamic WHERE clauses.

≡ Syntax

```
pushdown_sql (<datastore>, <input_string>)
```

Return Value

None.

Where

<code><datastore></code>	The name of the datastore containing the data you want to retrieve. Data Services creates a WHERE clause and pushes it down to this database. Surround the datastore name by single quotes.
<code><input_string></code>	A character string that forms the WHERE clause. Surround the character string by single quotes. Typically, this is a column from another source to the query, such as an XML message. Delimit columns or parameters with curly braces. For example, {XML_IN.STATUS_QUERY}. If the string contains a curly brace, use the backslash escape key to delimit the curly brace.

Details

The `pushdown_sql` function allows you to create WHERE clauses that change based on data input. With the `pushdown_sql` function, the WHERE clause need not be pre-specified. The `pushdown_sql` function is particularly useful in real-time jobs, if you want to select data based on input from an XML message.

Unlike other functions, the `pushdown_sql` function can only be used in the WHERE clause of a Query transform. You cannot use the `pushdown_sql` function in other places, such as in a query's mapping, in a conditional, or in a script.

Data Services must be able to push the WHERE clause that it creates from this function to the database. This function works best, therefore, when used in a Query transform where the immediate input is the table source where you want to push the WHERE clause.

Data Services does not parse the SQL contained in the input string. Therefore, the input must be well-formed with correct syntax.

i Note

Data Services does not allow use of the backslash escape key to delimit curly braces within the `pushdown_sql` function. So, if your input string contains a curly brace, you must make the string into a variable. Therefore, instead of entering 'a{b}c', you would pass your data through as 'a{\$x}c' where \$x = '{b}'.

❖ Example

Suppose the datastore EC_DS contains the table BIKES, which stores information about different models. And suppose the QUERY_REQUEST column in the XML_IN message contains requests for information from this table. For example, a value in the QUERY_REQUEST column might be:

```
TYPE = 'MOUNTAIN' and PRICE < 1500
```

In a data flow used in a real-time job, you can use the pushdown_sql function in a query to select data from the BIKES table based on the data in the XML_IN message. You can return the data in another XML message.

Function	Results
<pre>pushdown_sql ('EC_DS', '{XML_IN.QUERY_REQUEST}')</pre>	<p>Data Services includes the following WHERE clause in the SQL SELECT statement:</p> <pre>WHERE TYPE = 'MOUNTAIN' and PRICE < 1500</pre>

9.3.116 quarter

Determines the quarter in which the given date falls.

≡ Syntax

```
quarter(<date1>)
```

Return value

int

The number from 1 to 4 that represents the quarter component of <date1>.

Where

<date1>

The source date.

❖ Example

Function	Results
<code>quarter(to_date('Jan 22, 1997', 'mon dd, yyyy'))</code>	1
<code>quarter(to_date('5/97', 'mm/yy'))</code>	2

9.3.117 raise_exception

Calling this function causes an exception to be generated.

≡ Syntax

```
raise_exception(<error_msg>)
```

Return Value

int

Returns '1' always.

Where

<code><error_msg></code>	The string which will be written to the Job Server's error log.
--------------------------------	---

Details

The work flow or job may or may not be terminated based on whether a try-catch block surrounds the call.

❖ Example

```
ifthenelse(sal < 1000000, 0, raise_exception('Salary exceeds 1 million dollars.'))
```

9.3.118 raise_exception_ext

Calling this function causes an exception to be generated with an exit code.

Syntax

```
raise_exception_ext(<error_msg>, <exit_code>)
```

Return Value

int

Returns '1' always.

Where

<code><error_msg></code>	The string which will be written to the Job Server's error log.
<code><exit_code></code>	Code the job exits with, if the exception is not caught in a try/catch block. Use a number in range 1 to 255 (zero means "success" to all operating systems).

Details

The work flow or job may or may not be terminated based on whether a try-catch block surrounds the call.

Example

```
ifthenelse(sal < 1000000, 0, raise_exception_ext('Salary exceeds 1 million  
dollars.', sal/1000000 + 1))
```

9.3.119 rand

Returns a random number between 0 and 1.

Syntax

```
rand()
```

Return value

real

The random number. The return value is between 0 and 1.

Example

Function	Results
<code>100 * rand()</code>	A random number between 0 and 100.

9.3.120 rand_ext

Similar to, and more powerful than the `rand` function, `rand_ext` returns a random number between 0 inclusive and 1 exclusive. This function uses the linear-congruential generator (LCG) algorithm, $x_n = (ax_{n-1} + b) \bmod m$ where:

x_n is an integer from 0 to $m-1$ and the initial value of x_n is called the "seed" (x_0). For each call to the random number generator, the software calculates a new x_n by taking the value of the previous result x_{n-1} , multiplying by a , adding b , then taking the remainder mod m .

Data Services uses this formula to generate an integer from 0 to $m-1$. After Data Services calculates x_n , it divides that number by m to obtain a number equal to or greater than 0 and less than 1.

By specifying the same seed number, you can regenerate an exact number sequence (for use in repeat experiments).

Syntax

```
real rand_ext(<seed>)
```

Return value

real

The random number. The return value is between 0 and 1.

Where

<seed>

(Optional) Can be any positive integer greater than or equal to 0. If unspecified, the software uses the current time to create a seed.

❖ Example

Function	Results
<code>100 * rand_ext()</code>	A random number between 0 and 100.

9.3.121 regex_replace

Matches the whole input string to the pattern that is specified with regular expressions (regular expressions based on the POSIX standard) and flags and replaces the matching part of the input string with the replacement string provided.

≡ Syntax

```
regex_replace(<input string>, <regular expression pattern string>,  
<replacement string>, <regular expression processing flags>)
```

Return Value

String

Details

Matches the whole input string to the pattern that is specified with regular expressions (regular expressions based on the POSIX standard) and flags and replaces the matching part of the input string with the replacement string provided. The flags are optional.

❖ Example

Function	Results
<code>regex_replace('Accounting', '\\[a-h\\]', 'X', 'CASE_INSENSITIVE')</code>	'XXxountXnX'
<code>regex_replace('Accounting', '\\[a-h\\]', 'X')</code>	'AXXoutXnX'

Related Information

[match_regex \[page 1166\]](#)

9.3.122 replace_substr

Takes an input string, replaces each occurrence of a specified substring with a specified replacement, and returns the result.

≡ Syntax

```
replace_substr(<in_str>, <search_str>, <replace_str>)
```

Return Value

varchar

Where

<code><in_str></code>	The input string that you are changing. If NULL, return will be NULL.
<code><search_str></code>	String to search for. If <code><search_string></code> is NULL, varchar is returned and will be the same as <code>in_str</code> .
<code><replace_str></code>	If <code><replace_string></code> is omitted or NULL, all occurrences of <code><search_string></code> are removed.

❖ Example

Function	Result
<pre>replace_substr('a penny saved is a penny earned', 'penny', 'million')</pre>	<pre>'a million saved is a million earned'</pre>

9.3.123 replace_substr_ext

Takes an input string, replaces specified occurrences of a specified sub-string with a specified replacement and returns the result. It is also possible to search for the following:

- a hexadecimal value that refers to a UNICODE character
- a non-printable character reference such as a form feed or new line

≡ Syntax

```
replace_substr_ext(<in_str>, <search_str>, <replace_str>,  
<start_at_occurance>, <number_of_occurrences>)
```

Return Value

varchar

Where

<code>in_str</code>	The input string that you are changing. If NULL, returns NULL.
---------------------	--

`search_str`

String to search for:

You can use `/x0000` to specify the hexadecimal value for a special character. For example, if you use `/x000A`, then if Data Services encounters `/x` it will convert the next 4 characters to a hexadecimal value. This function converts the hexadecimal value to a UNICODE character. This option provides more flexibility when you use a search string.

You can also represent special characters using the escape character `'/'`. The following are supported.

`/a` Bell (alert)

`/b` Backspace

`/f` Formfeed

`/n` New line

`/r` Carriage return

`/t` Horizontal tab

`/v` Vertical tab

To include the escape character `'/'` in the search string, escape it using `'/'`. For example, if the input is `'abc/de'`, Data Services converts `search_str` to `'abcde'`. While if the input is `'abc//de'`, Data Services converts `search_str` to `'abc/de'`.

If `search_str` is NULL, Data Services returns a varchar with the data in `in_str`.

`replace_str`

String that replaces `search_str`. If `<replace_string>` is omitted or NULL, all occurrences of `<search_str>` are removed.

`start_at_occurrence`

Start replacing at this occurrence. If NULL, start at the 1st occurrence. For example, enter 2 to replace or remove the second occurrence of a `search_str`.

`number_of_occurrences`

Number of occurrences to replace. If NULL, replace all occurrences. For example, enter 2 to replace or remove two sequential occurrences of the `search_str`.

❖ Example

Function

Result

Replace 'a' with 'B' starting from second occurrence and replaces two occurrences:

`'ayyyByyyByyyayyy'`

```
replace_substr_ext('ayyyayyyayyyayyy', 'a', 'B',  
2, 2)
```

Search a string containing 'a' followed by a new line and replace it with 'B' starting from second occurrence and replaces two occurrences:

`'ayyyByyyByyyayyy'`

```
replace_substr_ext('ayyya</n>yyya</n>yyayyy', 'a/  
n', 'B', 2, 2)
```


Function	Result
Search a string containing 'a' followed by a new line and replace it with 'B' starting from second occurrence and replaces two occurrences: <code>replace_substr_ext('ayyya</n>yyya</n>yyyayyy', 'a/x000a', 'B', 2, 2)</code>	<code>'ayyyByyyByyyayyy'</code>

9.3.124 repository_name

Returns a database connection string and owner name. For example: `beq-local.DBUser`. This is the ID for the repository from which the job is run.

Syntax
<code>repository_name()</code>

Return Value

varchar

Example
<code>print('Repository Name: [repository_name()]')</code>

9.3.125 restore_repserver_cdb_backlogged_transactions

Restores backlogged transactions saved in the Replication Server real-time CDC datastore.

Syntax

<code>restore_repserver_cdb_backlogged_transactions (tablespec, dataflowname, jobname)</code>

where:

tablespect	<p><datastore>.<owner>.<table>. Datastore name must be a valid Replication Server real-time CDC data-store. <owner> and <table> can take a wild character *. Otherwise, the owner and table specified in the parameter must be imported to CDC datastore and executed to retrieve data from the queue.</p> <p>Examples:</p> <p>ORACLE_CDC_DATASTORE.PUBLISHER.BOOKS</p> <p>ORACLE_CDC_DATASTORE.*.BOOKS</p> <p>ORACLE_CDC_DATASTORE.PUBLISHER.*</p> <p>ORACLE_CDC_DATASTORE.*.*</p>
dataflowname	Valid data flow name or *
jobname	Valid job name

All parameters are mandatory and support global variables and substitution parameters.

Return values

0 Recovery request is already pending.

1 Recovery request has been recorded and will be processed by the CDC job when it finishes processing the current changed dataset transactions.

9.3.126 round

Rounds a given number to the specified precision.

Syntax

```
round(<num1>, <precision>)
```

Return value

decimal, double, int, or real

The rounded number. The return type is the same as the original number, <num1>.

Where

<code><num1></code>	The source number.
<code><precision></code>	An integer indicating the number of decimals in the result. If <code><precision></code> is negative, digits left of the decimal point are rounded.

❖ Example

Function	Results
<code>round(120.12345, 2)</code>	120.12
<code>round(120.12999, 2)</code>	120.13
<code>round(120, -2)</code>	100
<code>round(120.123, 5)</code>	120.12300

9.3.127 rpad

Pads a string with characters from a given pattern.

The function repeats the pattern at the end of the input string until the final string is the appropriate length. If the input string is already longer than the expected length, this function truncates the string.

≡ Syntax

```
rpad(<input_string>, <size>, '<pad_string>')
```

Return value

varchar

The new string.

Where

<code><input_string></code>	The source string.
-----------------------------------	--------------------

<code><size></code>	An integer value indicating the number of characters in the resulting string.
<code><pad_string></code>	A character or set of characters that this function concatenates to <code><input_string></code> .

❖ Example

Function	Results
<code>rpadd('Tanaka',15,' ')</code>	'Tanaka '
<code>rpadd(last_name,25,' ')</code>	The value in the column <code>last_name</code> , padded with spaces to 25 characters, or truncated to 25 characters.

9.3.128 rpadd_ext

Pads a string with logical characters from a given pattern.

i Note

These logical characters prohibit this function from getting pushed down to an Oracle database.

The function repeats the pattern at the end of the input string until the final string is the appropriate length. If the input string is already longer than the expected length, this function truncates the string.

≡ Syntax

```
rpadd_ext(<input_string>, <size>, '<pad_string>')
```

Return value

varchar

The new string.

Where

<code><input_string></code>	The source string.
-----------------------------------	--------------------

<code><size></code>	An integer value indicating the number of characters in the resulting string.
<code><pad_string></code>	A character or set of characters that this function concatenates to <code><input_string></code> .

❖ Example

Function	Results
<code>rpadd_ext('Tanaka',15,' ')</code>	'Tanaka '
<code>rpadd_ext(last_name,25,' ')</code>	The value in the column <code>last_name</code> , padded with spaces to 25 characters, or truncated to 25 characters.

`rpadd_ext` and `rpadd` functions exhibit the same behavior when the functions are evaluated. In situations where the function is pushed down to the database, the database behavior may differ when 'input_string' and/or 'pad_string' parameters contain multibyte characters.

Function	Input	Output
<code>rpadd</code>	<code>("abc",10,'')</code>	'abc '
<code>rpadd_ext</code>	<code>("abc",10,'')</code>	'abc '
<code>rpadd</code>	<code>("abc",10,"")</code>	'abc '
<code>rpadd_ext</code>	<code>("abc",10,"")</code>	'abc '
<code>rpadd</code>	<code>("abcd",10,'')</code>	'abcd '
<code>rpadd_ext</code>	<code>("abcd",10,'')</code>	'abcd '

9.3.129 rtrim

Removes specified characters from the end of a string.

≡ Syntax

```
rtrim('<input_string>', '<trim_string>')
```

Return value

varchar

The modified string. The return type is the same as `<input_string>`.

Where

<code><input_string></code>	The string to be modified.
<code><trim_string></code>	The characters to remove from <code><input_string></code> .

The function scans `<input_string>` right-to-left removing all characters that appear in `<trim_string>` until it reaches a character not in `<trim_string>`.

Removes trailing blanks only if `<trim_string>` contains trailing blanks. If the length of the modified string becomes zero after trimming, the function returns "" (empty string).

To remove all trailing blanks in a string, use the `rtrim_blanks` function.

❖ Example

Function	Results
<code>rtrim('Marilyn ', ' ')</code>	'Marilyn'
<code>rtrim('ZABCABC', 'ABC')</code>	'Z'
<code>rtrim('ZABCABC', 'EFG')</code>	'ZABCABC'

9.3.130 rtrim_blanks

Removes blank characters from the end of a string.

≡ Syntax

```
rtrim_blanks(<input_string>)
```

Return value

varchar

The modified string. The return type is the same as `<input_string>`.

Where

<code><input_string></code>	The string to be modified.
-----------------------------------	----------------------------

If the length of the modified string becomes zero after trimming, the function returns "" (empty string).

❖ Example

Function	Results
<code>rtrim_blanks('Marilyn ')</code>	'Marilyn'
<code>rtrim_blanks(last_name)</code>	The value contained in the column <code>last_name</code> with trailing blanks removed.

9.3.131 rtrim_blanks_ext

Removes blank and control characters from the end of a string.

≡ Syntax

```
rtrim_blanks_ext(<input_string>)
```

Return value

varchar

The modified string. The return type is the same as `<input_string>`.

Where

<code><input_string></code>	The string to be modified.
-----------------------------------	----------------------------

If the length of the modified string becomes zero after trimming, the function returns "" (empty string).

❖ Example

Function	Results
<code>rtrim_blanks('Marilyn ')</code>	'Marilyn'
<code>rtrim_blanks(last_name)</code>	The value contained in the column <code>last_name</code> with trailing blanks and control characters removed.

9.3.132 sap_openhub_processchain_execute

Performs the following tasks:

- Starts the process chain that extracts data from an InfoProvider (InfoArea, InfoCube, or DataStore object) on SAP NetWeaver Business Warehouse and loads the extracted data into an Open Hub Destination table.
- Monitors the process chain status and the Open Hub Destination request notification.

When the function returns successfully, an Open Hub table source in SAP Data Services can then read the data from the Open Hub Destination table.

i Note

You can only use this function in a script. It is not valid in a query or audit object.

Below is the function syntax as a reference. The function wizard is explained in the next section.

≡ Syntax

```
sap_openhub_processchain_execute('<datastore>', '<open_hub_table>',  
'<process_chain>', <$logid>, <$ReturnTxt>)
```

Where

<datastore>	<p>Specifies the datastore name. You can specify either a constant string or a substitution parameter.</p> <p>The data type is varchar(256).</p> <div><h3>i Note</h3><p>The maximum length depends on the Data Services repository type. For most repository types the maximum length is 256, for MySQL the length is 64, and for MS SQL server the length is 128.</p></div>
<open_hub_table>	<p>Specifies the Open Hub Destination table. You can specify either a constant string or a substitution parameter.</p> <p>The data type is varchar(30).</p>
<process_chain>	<p>Specifies the name of the process chain that extracts data from the InfoProvider in SAP NetWeaver BW and loads the data to the Open Hub Destination table. You can specify either a constant string or a substitution parameter.</p> <p>The data type is varchar(30).</p>
<\$logid >	<p>(Optional) Specifies a variable to obtain a value that depends on the function return value (see "Relationship between return value and value of logid variable" below).</p> <p>The required variable data type is varchar(25).</p>

<\$ReturnTxt>

(Optional) Specifies a variable to retrieve the description of the return status of the process chain.

The required variable data type is varchar, and you can define the length you want for this variable.

Return value

varchar(1)

Returns one of the following values.

Return value	Description
B	Open Hub Destination is being read by another user.
E	Data Services error while executing the function.
R	Process chain execution failed with errors in BW system.
X	Process chain execution has been canceled in BW system.
S	Function successfully executed the Open Hub extraction Data Transfer Process (DTP) and received extraction request notification.

Relationship between return value and value of logid variable

The value of the logid output variable depends on the function return value, as the following table shows.

Return value	\$logid variable value	\$ReturnTxt variable value	Action
B	Process chain log ID of the other user that is currently reading the Open Hub Destination	Status of current Open Hub extraction	Either wait and try again or stop executing the data flow that contains the Open Hub Destination table.
E	Data Services error log number	Data Services error text	Stop executing the data flow that contains the Open Hub Destination table, and analyze the Data Services error.
R	Your process chain log ID	Error from process chain	Stop executing the data flow that contains the Open Hub Destination table, and use the log ID in the BW system to see the detail state of the process chain error.
X	Your process chain log ID	Error from process chain	Stop executing the data flow that contains the Open Hub Destination table, and use the log ID in the BW system to see the detail state of the process chain error.

Return value	\$logid variable value	\$ReturnText variable value	Action
S	Open Hub extraction request ID	Status of your Open Hub extraction	Use the request ID in the BW system to obtain detail loading statistics (such as number of packets loaded and number of records loaded).

❖ Example

The following sample script commands check the return value, generate an exception, and print the error if the function is not successful.

```
$status = sap_openhub_processchain_execute('open_hub_datastore',
'Materials', 'Materials_PC', <$lpcogid>,
<$returntxt>);
If ($status != 'S') raise_exception ('Materials_PC process chain execution
failed ' || $returntxt);
```

Restrictions

The following are restrictions for using Open Hub Destinations:

- Only one job at a time can read an Open Hub Destination table.
- A process chain of an Open Hub Destination can contain only one of its Data Transfer Processes (DTPs).
- A process chain cannot contain DTPs for more than one Open Hub Destination.

9.3.132.1 Defining an sap_openhub_processchain_execute function

1. To access the function wizard for sap_openhub_processchain_execute from the Script Editor, click [Functions](#) or [...](#) at the top of the window.
2. Select [sap_openhub_processchain_execute](#) from the list of functions. The [Define Parameters](#) window opens.
3. Select an SAP BW Source datastore name from the drop-down list. You can also select a substitution variable from the list. If you type in a datastore name, put single quotes around the name.
4. Select the name of a Open Hub table from the drop-down list. Only the names of the imported Open Hub Tables appear in this list.
5. Select the name of a Process Chain from the drop-down list.
6. Specify a variable that will get the BW log ID for the process chain after the function executes. You must define the variable before you can use it.
7. Specify a variable that will get the description of the status after the function executes. You must define the variable before you can use it.

9.3.133 sap_openhub_set_read_status

Sends the read status for the Open Hub table to SAP NetWeaver BW. A successful read status causes SAP NetWeaver BW to delete the data from the Open Hub Destination table.

Syntax

```
sap_openhub_set_read_status('<datastore>', '<destination>',  
<status>, <$returntxt>)
```

Return value

varchar(1)

Returns one of the following values.

Return value	Description
S	Success
E	Error

Example

The following sample script commands sends the status of the Open Hub table read to the BW system and prints the status.

```
$status = sap_openhub_set_read_status('BR9", 'PS_BOOK_5', 'X', $returntxt):  
print ('Status: ' || $status);
```

Where

<datastore>

Specifies the datastore name. You can specify either a constant string or a substitution parameter.

The data type is varchar(256).

Note

The maximum length depends on the Data Services repository type. For most repository types the maximum length is 256, for MySQL the length is 64, and for MS SQL server the length is 128.

<code><destination></code>	<p>Specifies the Open Hub Destination in the BW system. You can specify either a constant string or a substitution parameter.</p> <p>The data type is varchar(30).</p>
<code><status></code>	<p>Specifies the read status. Possible values are either a variable or one of the following string constants:</p> <ul style="list-style-type: none"> • 'X' for Read Successful • Any other value indicates that the Read failed. <p>The data type is varchar(1).</p>
<code><\$returntxt></code>	<p>(Optional) Specifies a variable to return the status log of the function.</p> <p>The required variable data type is varchar, and you can define the length you want for this variable.</p>

9.3.134 search_replace

Performs a simple search and replace based on a string value, word value, or an entire field. You can specify search and replace values with an internal table, an existing external table or file, or with a custom SQL command. In all cases, the search and replace values are loaded into memory to optimize performance while performing the operation.

Note

We recommend that you use `search_replace` as a function call in the query output and not as a mapping in a column. By using it as a function call, you can use the wizard interface to fill in the function's parameters, and you can return to the wizard at any time to change the parameters. This method also allows you to create multiple output columns when multiple input expressions are used. Using the function in a script or regular mapping is possible, but the syntax can be hard to read and difficult to maintain.

Below is the function syntax as a reference. The function wizard is explained in the next section.

Syntax

```
search_replace([<sr_table_spec>,<search_column>,<replace_column>],<sr_type>,<case_sensitivity>],[<default_replace_value>],[<input_column_list>],[<output_column_list>],[<output_variable_list>]) SET (...)
```

Return value

varchar

❖ Example

Search for Mr in input_column and replace with M to output_column using an internal search and replace table.

```
search_replace(NULL, 'SR_STRING', 1, , 'input_column', 'output_column',) SET (
  "internal_table"='<?xml version="1.0" encoding="UTF-8"?>
    <searchTable>
      <item>
        <Search>Mr</Search>
        <Replace>M</Replace>
      </item>
    </searchTable>
  )
```

i Note

The search_replace function wizard makes it easy to select search and replace columns, and if needed, define search terms and replacement values.

Where

<code><sr_table_spec></code>	<p>A constant string that specifies the search and replace table or file. It has three possible valid forms:</p> <p><code><Datastore.owner.table></code> — Specifies a database table containing the search and replace values.</p> <p><code><Datastore.owner.schema.table></code> — Specifies a database table containing the search and replace values for Netezza 7.x multi-schema only .</p> <p><code><Fileformat.filename></code> — Specifies a fixed or delimited file containing the search and replace values.</p> <p>NULL — Used when the search and replace is performed using custom SQL or an internal table defined in the SET options.</p>
<code><search_column></code>	<p>Specifies the column name in the table or file containing the search values. If <code><sr_table_spec></code> is an internal table, this should be set to NULL.</p> <p>The data contained in this column must be of the varchar type.</p>
<code><replace_column></code>	<p>Specifies the column name in the <code><sr_table_spec></code> table or file containing the replacement values. If <code><sr_table_spec></code> is an internal table, this should be set to NULL.</p> <p>All data contained in this column must be of the varchar type.</p>

<code><sr_type></code>	<p>A string that specifies the type of search and replace operation to perform. It has three possible values:</p> <p>'SR_FIELD' — Matches the entire contents of the search field, and replaces the entire contents of the search field.</p> <p>'SR_WORD' — Replaces only the word that matches the search value. Any unmatched data remains in the search field. A word is defined as a set of characters set apart by whitespace.</p> <p>'SR_STRING' — Replaces only a specific substring of characters found next to or between other characters in the search field. Any unmatched data remains in the search field.</p>
<code><case_sensitivity></code>	<p>A value that indicates whether or not the search and replace operation is case sensitive.</p> <p>yes — case sensitive</p> <p>no — not case sensitive</p>
<code><default_replace_value></code>	<p>A varchar that specifies the default replacement value if the search value is not found.</p> <p>This applies only when <code><sr_type></code> is set to 'SR_FIELD'.</p>
<code><input_column_list></code>	<p>A varchar that specifies a comma-separated list of input expressions on which the search and replace operation should be performed.</p>
<code><output_column_list></code>	<p>A varchar that specifies a comma-separated list of output columns.</p>
<code><output_variable_list></code>	<p>A varchar that specifies a comma-separated list of output variables.</p> <p>When more than one output column is specified in <code><output_column_list></code>, the output variables are used to receive output returns. Variables and output columns are matched by position.</p> <p>This parameter is optional except when more than one output column is specified in <code><output_column_list></code>.</p>

SET options

Specifies custom SQL or search and replace values in XML format.

For an internal search and replace table:

```
SET (
  "internal_table"=
  '<?xml version="1.0" encoding="UTF-8"?>
    <searchTable>
      <item>
        <Search><value></Search>
        <Replace><value></Replace>
      </item>
      ...
    </searchTable>
  ')
```

For custom SQL:

```
SET (
  "external_custom_sql"=
  '<?xml version="1.0" encoding="UTF-8"?>
    <database_datastore>
      <datastore_name>
    </database_datastore>
    <search_column>
      <search_column_name>
    </search_column>
    <replace_column>
      <replace_column_name>
    </replace_column>
    <SQLText>
      <custom_sql_string>
    </SQLText>
  ')
```

9.3.134.1 Defining a search_replace function

1. Right-click the output schema of a Query transform, and click *New Function Call*.

2. Choose the *String Functions* category, and click *search_replace*.

3. Click *Next*.
The search_replace function wizard opens.

4. Select the search type to perform.

Search type	Description
Full string	Matches and replaces the entire contents of the input expression.
Word	Replaces only the word that matches the search value. Any unmatched data is not modified. A word is defined as a set of characters set apart by whitespace.
Substring	Replaces only a specific substring of characters found next to or between other characters in the input expression. Any unmatched data is not modified. For a substring search, the search values are ordered by length from largest to smallest, and each substring is only substituted once.

5. If you want to ignore casing differences between the specified search values and the values from the input expressions, uncheck *Case sensitive*.

6. Define the input expressions to search. You can select a column name from the input schema, or you can use any expression that uses one or more input columns.
7. Define the output column names and lengths for the replacement values. You can rename the output column by selecting the value and pressing **F2**.
8. Configure the search and replace table.
 - For an internal table, select *Internal* and specify search values and corresponding replacement values. The values for an internal table are stored in the metadata repository as part of the function definition.
 - For an external relational table, select *External*. Specify a valid datastore or file format as the Source, and select the columns containing the search and replacement values.
 - For custom SQL, select *Custom SQL*. Choose the source datastore, and specify the columns containing the search and replacement values. Define the custom SQL text to run.
9. Specify the default replacement value for any rows that do not match the defined search values. If the replacement value is a fixed string, you must enclose it in single quotes.

Note

The default replacement value applies only for a full string search, not for word or substring searches.

If no default replacement value is defined, the original values are preserved; no replacements are made for expressions that were not found.

10. If you want to force the search_replace function to execute in a separate process, check *Run as a separate process*. The search and replace table is always loaded into memory, so for large tables, running this function as a separate process can improve performance.

Example

Case-sensitive substring search and replace

In this example, the longest search value is replaced first. As a result, the substring AUS does not become AUSA, but is correctly replaced with Australia.

Internal Search and Replace table:

Search value	Replace value
US	USA
AUS	Australia

Search and Replace results:

Value on input	Value on output
California, US	California, USA
Melbourne, AUS	Melbourne, Australia

9.3.135 set_cdc_checkpoint

Sets a check-point in a data flow for a Microsoft SQL Server changed-data-capture (CDC method) job. Use for data flows that run in a WHILE loop to retrieve changed data for each iteration of the loop. Call this function for all the datastores used in all the data flows of the job.

Syntax

```
set_cdc_checkpoint(<datastore>)
```

Return value

int

Returns 1 if successful, otherwise 0.

Where

<code><datastore></code>	The name of the CDC-enabled datastore containing the tables used to obtain the changes.
--------------------------------	---

Example

```
set_cdc_checkpoint('MyCdcSource');
```

9.3.136 set_env

Sets a system environment variable to a specified value for the duration of a job.

Syntax

```
set_env('<variable_name>', <variable_value>)
```

Return value

int

Returns 1 if successful, otherwise, 0.

Where

<code><variable_name></code>	The name of an environment variable. The name must be surrounded by single quotes.
<code><variable_value></code>	The value you want assigned to the environment variable. If the value is text, you must surround the text by single quotes. Data Services only sets the variable to this value for the duration of the current job.

Use the `get_env` and `set_env` functions to set and retrieve variables across operations in a job.

❖ Example

Function	Results
<pre>set_env('TMP', 'C:\Data Services\Temp')</pre>	Sets the value of the TMP environment variable to "C:\Data Services\Temp" and returns a 1.

Related Information

[get_env \[page 1104\]](#)

9.3.137 sleep

Suspends the execution of the calling data flow or work flow.

≡ Syntax

```
sleep(<num_millisecs>)
```

Return Value

int

Returns '1', always.

Where

<num_millisecs>

The number of milliseconds to "sleep".

Details

Calling this function causes the thread that executes this function to halt operations for the given number of milliseconds. To force a job to halt operations (until a condition becomes true), call this function in a work flow, not in a data flow.

❖ Example

The following example invokes sleep for one second when a file exists in a directory called 'c'.

```
while (file_exists('c:/temp.msg') == 0)
begin
sleep(1000);
end
```

9.3.138 soundex

Encodes the input string using the soundex algorithm and returns a string. Use this function when you want to push-down to the database-level. Results may vary when you push-down to different database types.

≡ Syntax

```
soundex(<input_str>)
```

Return Value

varchar(4)

Returns a string containing the soundex encoding of the input string. The return string length is always four characters.

Where

`<input_str>`

The source string that will be encoded.

Details

Only use this function for input strings in English. Non-English characters are ignored.

Any invalid leading characters in the input string are ignored.

If an input string cannot be encoded, then '0000' is returned.

❖ Example

Function	Result
<code>Print (soundex ('Hello')) ;</code>	Prints the soundex of the word "Hello."
<code>\$VAR=soundex (emp_name) ;</code>	Returns the soundex encoding for the string stored in the variable emp_name and then assigns it to \$VAR
<code>\$VAR=soundex ('1234567') ;</code>	Returns '0000' because the input data is numeric.

9.3.139 sql

Runs a SQL operation against tables in the specified database.

≡ Syntax

```
sql (<datastore, sql_command>)
```

Return value

varchar(1020)

Returns the first 1020 characters from the query's output. Typically, if `<sql_command>` is a SELECT statement, the return value is the first row value of the first column. If `<sql_command>` is not a SELECT statement, the return value is typically NULL. You must remember this if you assign the value returned to a variable.

Where

`<datastore>` A string containing the name of the datastore where the tables involved in the SQL operation reside. This name is the name you specified when you created the datastore in Data Services. Include this string in single quotation marks.

`<sql_command>` A string containing the text of the SQL command to execute. This string must be enclosed in single quotation marks ('). If the string contains quoted values, the internal quotation marks must be single quotation marks preceded by the escape character, backslash (\).

Data Services makes column and table names uppercase when sending the `<sql_command>` to Oracle to resolve. To specify a lowercase column or table name from an Oracle database, enclose the name with double quotation marks (").

❖ Example

Function	Results
<pre>sql('source_ds', 'SELECT EmpID FROM Emp WHERE sal > 100000')</pre>	Runs the SQL command against the database connected through the datastore <code>source_ds</code> .
<pre>sql('source_ds','SELECT customer.LastName FROM customer WHERE customer.State = \'CA\')</pre>	Returns the last names of the customers in California from the customer table. Note that the quotation marks around the state value require a backslash to indicate that the single quotation marks are considered part of the <code><sql_command></code> string.
<pre>sql('oracle_ds', ('SELECT "start_timestamp" FROM "status_table" WHERE "extract_name" = 'DF RecoverDim' AND "stop_timestamp" = NULL'))</pre>	Returns the timestamp from a status table for the completed data flow. An Oracle datastore requires double quotation marks around the lowercase column and table names.
<pre>\$start_date=sql('warehouse_ds', 'SELECT finish_timestamp FROM time_table WHERE table_name= "Component_Orders" ');</pre>	In this script example, because the function returns a varchar value, it is not possible to assign the return value to a date variable directly. Modify your statement to the next example.
<pre>\$temp_char=sql('warehouse_ds', 'SELECT finish_timestamp FROM time_table WHERE table_name= "Component_Orders" '); \$start_date=to_date(\$temp_char,'dd- mon-yyyy');</pre>	This script example assumes the database returns the date in dd-mon-yyyy format. If you are unsure of the format the database returns, then force it to return the date in a specific format by doing a conversion. To accomplish this, use the <code>to_char</code> function in Oracle or the <code>convert</code> function in MS SQL.

9.3.140 sqrt

Returns the square root of the given expression.

Syntax

```
sqrt (<num>)
```

Return Value

Float

Where

<num>

The number for which you want the square root.

Details

Return value is NULL if the input is negative.

Example

Function	Results
sqrt (625.25) ;	25.005000

9.3.141 smtp_to

Captures the specified number of lines in the trace and error logs, packages the information into an e-mail, and sends it to the recipient(s) via an SMTP server. This function is typically used in a script, for example in a conditional clause, while loop, or try-catch block.

Syntax

```
smtp_to (<recipients_list>, <subject>, <message>, <number_of_trace_lines>,  
<number_of_error_lines>)
```

Return value

int

Returns 0 if function succeeds. Returns a non-zero integer if function fails.

Where

<code><recipients_list></code>	A string containing one or more recipient e-mail addresses separated by commas (.). This string cannot be empty and must contain valid, qualified e-mail address information.
<code><subject></code>	A string containing the subject of the e-mail. This string can be empty.
<code><message></code>	A string containing the e-mail message. This string can be empty.
<code><number_of_trace></code> <code><_lines></code>	The number of lines from the end of the trace log file to append to the end of the e-mail. This input cannot be empty.
<code><number_of_error></code> <code><_lines></code>	The number of lines from the end of the error log file to append at the end of the e-mail. This input cannot be empty.

❖ Example

smtp_to function

Function	Results
<pre>\$myvar = smtp_to('admin@company.com', 'Out of memory error in the SalesFact job. Please fix the error before running recovery job.', ' ', 10, 10)</pre>	<p>The message goes to one recipient, admin@company.com.</p>
<pre>\$myvar = smtp_to('admin@company.com, manager@company.com', 'Out of memory error:' systime(), 'Out of memory error while running the data flow:' \$dataflow_name ' in the job:' \$job_name '.', 10, 20)</pre>	<p>The message goes to two recipients, admin@company.com and manager@company.com.</p> <p>The job name and data flow names are included in the text of the message as variables. Note that Data Services trims blank spaces from the end of strings; this example includes a blank on the beginning of the next string. You can also concatenate a string with a single blank.</p> <p>In the script, type <code>\$a = ;</code> where <code>\$a</code> is the local integer variable defined in the work flow. Put the cursor just ahead of the semicolon before clicking the Functions button to construct a <code>smtp_to</code> statement.</p>

Function	Results
<pre> \$address_list = 'nkumar@businessobjects.com,pkulkarn@ businessobjects.com'; \$subject_text = 'Testing SMTP job smtp011'; \$message_text = 'The job smtp011 has the following trace lines & errors'; \$trace_log = 8; \$error_log = 19; \$smtp_output = 9; print ('before execution :- ' \$smtp_output); \$smtp_output = smtp_to(\$address_list, \$subject_text,\$message_text, \$trace_log,\$error_log); print ('after execution :- ' \$smtp_output); </pre>	<p>The smtp_to function also supports global variables. This example from a script shows the smtp_to function substituting the values from global variables.</p>
<p>Create a custom function my_smtp that contains the following:</p> <pre> begin return(smtp_to('nkumar@ness- gsg.com','Testing job smtp012','Test message of smtp012 job',4,5)); End </pre> <p>Use the custom function my_smtp in a script as follows:</p> <pre> print('before smtp :- '); my_smtp(); print('after smtp :- '); </pre>	<p>You can use smtp_to in any user-defined custom function and invoke it in a script.</p>
<p>i Note</p> <p>The smtp_to function does not support nicknames.</p>	

9.3.141.1 Defining and enabling the smtp_to function

1. In the function editor, click the [System](#) Function Category.
2. Click the [smtp_to](#) function name.
3. Click [Next](#).
4. Define the input parameters as described in the [Where](#) table previously in this function description.
5. Click [Finish](#).
6. Open the Data Services Server Manager:
 - In Windows, click **Start** > **Programs** > **SAP Data Services <x.x>** > **Data Services Server Manager**.
 - In UNIX, run the Server Manager by entering:

```
$ cd $LINK_DIR/bin/
```



```
$ . ./al_env.sh
```

```
$ ./svrcfg
```

7. The Data Services Server Manager Utility window opens.
8. Enter the SMTP server settings:
 - **Server name**: Type the name or IP address of the SMTP server (for example mail.company.com)
 - **Sender**: Type the e-mail address that will appear in the *From* field of the e-mail.
9. In the Server Manager window, click *Apply*.

9.3.142 string_to_number

Returns the integer sum of all characters from the input string.

Syntax

```
string_to_number(<input string>)
```

Return Value

Int

Details

Returns the integer sum of all characters from the input string. Input string size is limited to 4000 characters.

Example

Function	Results
<code>string_to_number('SAP')</code>	228

9.3.143 substr

Returns a specific portion of a string starting at a given point in the string.

Syntax

```
substr(<input_string>, <start>, <length>)
```

Return value

varchar

The modified string. The return data type is the `<input_string>`. If the length is a constant, then it is a varchar of the given length.

Where

<code><input_string></code>	The string to be modified.
<code><start></code>	<p>The position of the first character in the new string. The function counts characters from the beginning of <code><input_string></code>. In normal data flows, the first character is position number 1. If <code><start></code> is 0, the new string begins with the first character (position 1).</p> <p>If <code><start></code> is negative, the function counts characters from the end of <code><input_string></code>. The new string begins with the character in that position from the end of the string.</p> <p>If <code><start></code> is greater than the number of characters in <code><input_string></code>, the function returns NULL. If <code><length></code> is 0 or negative and <code><start></code> is not greater than the number of characters in <code><input_string></code>, the function returns empty string.</p>
<code><length></code>	<p>The number of characters in the resulting string. If <code><length ></code> is 0 or negative, the function returns an empty string. If <code><length></code> is greater than the number of characters remaining in <code><input_string></code> after <code><start></code>, the function returns only the remaining characters.</p> <p>The function keeps the trailing blanks in the remaining <code><input_string></code> after <code><start></code>.</p>

Example

Function	Results
<code>substr('94025-3373', 1, 5)</code>	'94025'
<code>substr('94025-3373', 7, 4)</code>	'3373'
<code>substr('94025', 7, 4)</code>	NULL

Function	Results
<code>substr('Dr. Schultz', 4, 18)</code>	'Schultz'
<code>substr('San Francisco, CA', -4, 18)</code>	', CA'

9.3.144 sum

Calculates the sum of a given set of values.

Syntax

```
sum(<value_list>)
```

Return value

decimal, double, int, or real

The total of the values. The return type is the same as the values in `<value_list>`.

Where

`<value_list>`

The source values to sum.

Example

To calculate the sum of values in the salary column of a table, use the sum function in a query:

- In the *Mapping* tab of the query editor, enter:

```
sum (SALARY)
```

- In the *Group By* tab in the query editor, specify the columns for which you want to find the total salary, such as the department column. For each unique set of values in the group by list, such as each unique department, Data Services calculates the sum of the salary.

9.3.145 sysdate

Returns the current date as listed by the Job Server's operating system.

Note

The value that the `sysdate` function returns is a `datetime` value. Internally Data Services reads both the date and the time when it runs a `sysdate` function. The data that is used by the job depends on the data type of a particular column. For example, if the data type of a column in a query is `date`, Data Services only uses the date for calculations. The time data is ignored. If you change the data type to `datetime`, both a date and a time are used.

Syntax

```
sysdate()
```

Return value

`date`

Today's date.

Example

Function	Results
<code>isweekend(sysdate())</code>	Tests whether today is a Saturday or Sunday.
<code>to_char(sysdate(), 'yyyy.mm.dd')</code>	<p>Converts the <code>sysdate</code> function's <code>datetime</code> value to a string that displays only the date.</p> <p>For example, you can use this to exclude part of the <code>datetime</code> data by only providing a format for the data you want to display in a report. To convert a <code>datetime</code> value to a string containing only the date, use this expression and change the column's data type to <code>varchar</code>.</p>

9.3.146 system_user_name

Returns the user name used to log into the Job Server's operating system.

Syntax

```
system_user_name()
```

Return Value

varchar

❖ Example

```
print('Starting execution of Job: [job_name()] as user: [system_user_name()]');
```

9.3.147 systime

Returns the current time as listed by the Job Server's operating system.

≡ Syntax

```
systime()
```

Return value

time

The current time.

❖ Example

Function	Results
<pre>\$timestamp = sql('my_datastore', ('UPDATE status_table SET job_start_time = \' ' to_char(systime(), 'hh24:mi:ss.ff') '\' ');</pre>	This expression updates the <code>job_start_time</code> column of the <code>status_table</code> with the current time. It also formats the time data.

Function	Results
<code>to_char(sysdate(), 'hh24:mi:ss.ff')</code>	<p>Trims date data from the <code>sysdate()</code> function in cases where it is added by default. Set the column that contains this expression to the data type <code>varchar</code>.</p> <p>The data type for a column that calls the <code>sysdate()</code> function should be <code>time</code>. If the data type is set to <code>datetime</code>, Data Services will add the default date for the <code>datetime</code> data type (1900:01:01) because <code>sysdate()</code> does not read dates.</p>

9.3.148 table_attribute

Retrieves the value of the specified table attribute.

Syntax

```
table_attribute(<table_name, attribute_name>)
```

Return Value

varchar

The value of the table attribute. If the specified attribute does not exist, NULL is returned.

where

`<table_name>`

Use the following format: 'datastore.owner.table' or 'datastore.owner.schema.table' (Only for Netezza 7.x multi-schema). If a valid table by this name does not exist, NULL is returned. This parameter is case sensitive.

`<attribute_name>`

The name of a table attribute. Valid attributes for a table are listed in the table's Properties window. This parameter is case sensitive.

❖ Example

Function	Result
<code>table_attribute('mssql.awez.CUSTOMER', '1788' 'Number_Of_Inserts')</code>	

9.3.149 to_char

Converts a date or numeric data to a string. It supports the Oracle 9i `timestamp` data type up to 9 digits precision for sub-seconds.

≡ Syntax

```
to_char(<date or numeric_expression>, '<format>')
```

Where

A formatted string describing `<numeric_expression>`.

<code><numeric_expression></code>	The source int, real, double or decimal value.
<code><format></code>	A string indicating the format of the generated string.

i Note

Please provide format to ensure correct results.

Choose from the following codes:

Format	Description	Example
9	Number (suppress leading/trailing zeros) Includes a leading - for negative numbers or one leading space for pos no's.	<code>to_char(123,'9999') = '123'</code>
0	Number Including leading/trailing zeros.	<code>to_char(123,'09999') = '0123'</code> <code>to_char(123,'9999D.00') = '123.00'</code>

Format	Description	Example
D<. ,>	Position of decimal point followed by character to be used as decimal separator. Currently only dot(.) and comma(,) are supported as decimal points. Currently only dot(.) and comma(,) are supported as decimal points.	to_char(12.34,'99D.99') = '12.34'
G<. , space >	Position of group separator followed by character to be used as group separator. Currently only dot(.), comma(,) and space(' ') are supported as group separator.	to_char(1234,'9G,999') = '1,234'
x	String containing unsigned hexadecimal integer, using "abcdef". Output is not padded if number is not 2 bytes long.	to_char(123,'xx') = '7b' to_char(12,'x') = 'c'
X	String containing unsigned hexadecimal integer, using "ABCDEF". Output is not padded if number is not 2 byte long.	to_char(123,'XX') = '7B' to_char(12,'X') = 'C'
O	String containing unsigned octal integer. This option is "not" case sensitive. Output is not padded if number is not 2 bytes long.	to_char(12,'oo') = '14' to_char(1,'o') = '1'
A formatted string describing <date>.		
<date>	The source date, time, or datetime value.	

`<format>`

A string indicating the format of the generated string. Choose from the following codes:

DD: 2-digit day of the month

MM: 2-digit month

MONTH: Full name of month

MON: 3-character name of month

YY: 2-digit year

YYYY: 4-digit year

HH24: 2-digit hour of the day (00-23)

MI: 2-digit minute (00-59)

SS: 2-digit second (00-59)

FF: Up to 9-digit sub-seconds

Other values included in `<format>` appear in the result unchanged.

Return value

varchar

❖ Example

Function	Results
<code>to_char(call_date, 'dd-mon-yy hh24:mi:ss.ff')</code>	The date value from the <code>call_date</code> column formatted as a string, such as: 28-FEB-97 13:45:23.32

The hyphens and spaces in `<format>` in the example are reproduced in the result; all the other characters are recognized as part of a parameter string in the table above and substituted with appropriate current values.

Related Information

[timestamp \[page 315\]](#)

9.3.150 to_date

The `to_date` function converts a string to a date based on the input format.

If the input string has more characters than the format string, the extra characters in the input string will be ignored and will be initialized to the default value.

For example, `to_date('10.02.2007 13:25:45', 'DD.MM.YYYY')` will convert the date to 10.02.2007 00.00.00. The time part in the input string will be ignored and initialized to 0.

This function also supports the Oracle 9i `timestamp` data type. Its precision allows up to 9 digits for sub-seconds.

Syntax

```
to_date (<input_string>, '<format>')
```

Return value

date, time, or datetime

A date, time, or both representing the original string.

Where

<input_string>	The source string.
<format>	A string indicating the format of the source string. Choose from the following codes: <div>DD: 2-digit day of the month</div> <div>MM: 2-digit month</div> <div>MONTH: Full name of month</div> <div>MON: 3-character name of month</div> <div>YY: 2-digit year</div> <div>YYYY: 4-digit year</div> <div>HH24: 2-digit hour of the day (00-23)</div> <div>MI: 2-digit minute (00-59)</div> <div>SS: 2-digit second (00-59)</div> <div>FF: Up to 9-digit sub-seconds</div>

❖ Example

Function	Results
<code>to_date('Jan 8, 1968', 'mon dd, yyyy')</code>	1968.01.08 stored as a date.

Related Information

[timestamp](#) [page 315]

9.3.151 to_decimal

Converts a varchar to a decimal.

≡ Syntax

```
to_decimal('<in_str>', '<decimal_sep>', '<thousand_sep>', <scale>)
```

Return Value

decimal

Uses a precision of 28 and the given scale.

Where

<code><in_str></code>	The number string. Null implies a NULL return.
<code><decimal_sep></code>	The character that separates the decimal component from the whole number component.
<code><thousand_sep></code>	The character that separates thousands from hundreds in the whole number component.

<code><scale></code>	The number of digits to the right of the decimal point in the returned value.
----------------------------	---

Details

Takes a string that represents a number and converts it to a decimal. If the input string is invalid, a 0 is returned.

Example

Function	Result
<code>to_decimal('99,567.99', '.', ',', 3)</code>	99567.990

9.3.152 to_decimal_ext

The `to_decimal_ext` function supports the use of DECIMAL data types with up to 96 precision. This function converts a varchar to a decimal and includes precision as a parameter.

Syntax

```
to_decimal_ext(<'in_str'>, <'decimal_sep'>, <'thousand_sep'>, <precision>, <scale>)
```

Return Value

decimal
Uses the given precision and scale.

Where

<code><in_str></code>	The number string. Null implies a NULL return.
-----------------------------	--

<decimal_sep>	The character that separates the decimal component from the whole number component.
<thousand_sep>	The character that separates thousands from hundreds in the whole number component.
<precision>	The total number of digits in the returned value.
<scale>	The number of digits to the right of the decimal point in the returned value.

Details

Takes a string that represents a number and converts it to a decimal. Returns 0 if the input string is invalid.

❖ Example

Function	Result
<code>to_decimal_ext('99,567.99', '.', ',', 38, 3)</code>	99567.990

9.3.153 to_WKT_point

Converts latitude and longitude to a geometry point in Well Known Text (WKT) format.

The software outputs point data using geometry data type. You can upload the data objects to your external spatial data, such as a shapefile. This function is limited to loading data to SAP HANA.

The return value represents a point on the map based on the decimal latitude and longitude from your input data. The software outputs the WKT point information as long data type. Long data type provides enough space so that the WKT does not get truncated.

≡ Syntax

≡ Sample Code

```
to_WKT_point (x, y)
```

Return value

Long

Point data that represents the location on a map for the given latitude and longitude.

Where

<x>	Latitude
<y>	Longitude

You access the function using the Select Function feature in a Query transform.

1. In *Select Function*, click *Conversion Functions* from the Function Categories list.
2. Select *to_WKT_point* from the Function Names list and click *Next*.
3. In the *Define Input Parameter(s)* view, select the latitude field from the *X Coordinate* dropdown list.
4. Select the longitude field from the *Y Coordinate* dropdown list.
5. Click *Finish*.

The software issues an error if the input latitude and longitude fields are not in one of the following formats:

- Decimal
- Integer
- Float
- Double

❖ Example

Input:

x = 3

y = 6

Output:

POINT (3 6)

9.3.154 total_rows

Returns the number of rows in a particular table in a datastore. This function can be used with any type of datastore.

≡ Syntax

```
total_rows(<datastore.owner.table_name>)
```

or for a Netezza 7.x multi-schema datastore:

```
total_rows(<datastore.owner.schema.table_name>)
```

or for a memory datastore:

```
total_rows(<datastore..table_name>)
```

Return value

int

The number of rows in the table.

Where

<code><datastore></code>	The name of the datastore containing the table.
<code><owner></code>	The name of the datastore owner. Not used for memory tables.
<code><schema></code>	The name of the datastore schema. Only used for Netezza 7.x datastore.
<code><table_name></code>	The name of the database table or memory table containing the rows you want to count.

❖ Example

Function	Results
<code>total_rows (ora_ds.scott.emp_table)</code>	Retrieves the total number of rows from an Oracle table.
<code>total_rows (mem_ds..bigtable)</code>	Retrieves the total number of rows from a memory table.

9.3.155 translate

Takes the input string and translates each character to its corresponding mapping to return the output string.

Syntax

```
translate(<input string>, <from string>, <to string>)
```

Return Value

String

Details

Returns the translated input string with all occurrences of each character in the `<from string>` replaced by its corresponding character in the `<to string>`. Characters in the input string that are not in the `<from string>` are not replaced. The characters in the `<from string>` that do not have a corresponding `<to string>` are removed from the input string. If the `<from string>` or `<to string>` is null, then it returns null. This function is case sensitive with parameter values.

Example

Function	Results
<code>translate('Business Objects','sne','xyz')</code>	'Buxiyzxx Objctx'
<code>translate('Business Objects','st','x')</code>	'Buxinexx Objecx'
<code>translate('Business Objects','s','')</code>	'Buine Object'
<code>translate('Business Objects','abcd',NULL)</code>	NULL

9.3.156 trunc

Truncates a given number to the specified precision, without rounding the value.

Syntax

```
trunc(<num1>, <precision>)
```


Return value

decimal, double, int, or real

The truncated number. The return type is the same as the original number, `<num1>`.

Where

<code><num1></code>	The source number.
<code><precision></code>	An integer indicating the number of decimals in the result. If <code><precision></code> is negative, digits to the left of the decimal point are truncated and the value is padded with zeros.

❖ Example

Function	Results
<code>trunc(120.12345, 2)</code>	120.12
<code>trunc(120.12999, 2)</code>	120.12
<code>trunc(180, -2)</code>	100
<code>trunc(120.123, 5)</code>	120.12300

9.3.157 truncate_table

Allows you to explicitly expunge data from a memory table or truncate physical files used for a persistent cache table. With regard to memory tables, this function provides finer control than the active job has over your data and memory usage. Use this function with memory tables and persistent cache tables.

≡ Syntax

```
trunc(<ds..tab_name>)
```

Return value

int

The return value is always 1.

Where

<code><ds></code>	The datastore containing the memory table or persistent cache table.
<code><tab_name></code>	The name of the memory table from which you want to expunge data or the name of the persistent cache table from which you want to truncate physical files.

❖ Example

Function	Results
<code>truncate_table (ds..bigtable)</code>	Truncates rows from the memory table or persistent_cache table.

Usage scenarios

1. A data flow in your job creates a persistent cache table which can be used by subsequent data flows within the job (through the `lookup_ext` function, for example).
You can add a custom function directing the `truncate_table()` function to remove the persistent cache table after running all data flows within the job.
2. Create a job that includes a script to clean up all unused persistent cache tables. When run, the job would truncate your physical files and free disk space.

9.3.158 upper

Changes the characters in a string to uppercase.

≡ Syntax

```
upper (<value>, '<locale>')
```

Return value

varchar

The uppercase string. The return type is the same as `<value>`. Any characters that are not letters are left unchanged.

Where

<code><value></code>	The string to be modified.
<code><locale></code>	Optional parameter that converts the string to the specified locale.

Note
ISO 639 language code and ISO 3166 country code formats are supported.

Example

Function	Results
<code>upper('Accounting101')</code>	'ACCOUNTING101'
<code>upper(substr(LastName,1,1)) lower(substr(LastName, 2,LENGTH(LastName)))</code>	The value in column <code>LastName</code> with the first letter uppercase and the rest of the value lowercase. Note that this example does not account for two-word last names.
<code>upper(LastName, 'tr')</code>	The value in column <code>LastName</code> is converted to all lowercase. It is also converted to the Turkish locale, using the ISO 639 language code.

Related Information

[ISO 639 language list](#) 📄

[ISO 3166 Country Code list](#) 📄

9.3.159 utc_to_local

Converts the input in Coordinated Universal Time (UTC) to the desired time zone value.

Syntax

```
utc_to_local(<input datetime>, <timezone to convert with UTC offset>)
```

Return Value

datetime

Details

Converts the input in Coordinated Universal Time (UTC) to the desired time zone value. The second parameter UTC offset is a constant value. If the UTC offset is not provided, then it is taken as the time zone of the job server host to calculate the UTC offset.

Example

Function	Results
<pre>utc_to_local('2014.01.31 15:30:00', 'UTC+08:30')</pre>	<pre>'2014.02.01 00:00:00'</pre>

9.3.160 varchar_to_long

Converts a data type value of a given column from varchar to long.

Syntax

```
varchar_to_long(<column_name>)
```

Return value

long

Where

<code><column_name></code>	The name of the table column for which you want to convert a data type from varchar to long.
----------------------------------	--

9.3.161 wait_for_file

This function looks for the specified file pattern in the file system. If it does not find the file(s), it waits for the specified timeout period, polling for the file(s) at every polling interval. The value specified in poll_interval determines how often to poll for the file pattern until timeout is reached. After timeout, the job stops, and polling for the file ceases.

Syntax

```
wait_for_file ( file_name_pattern, timeout, poll_interval,max_match, file_name_list, list_size, list_separator)
```

Return Values

int

Values are:

- 0 - No file matched.
- 1 - At least one file is matched.
- -1 - Timed out.
- -2 - At least one input value is illegal.

Where

<code><file_name_pattern></code>	The file name and path, relative to where the Job Server is running. It can be an absolute or relative path. File name can contain wildcard characters.
--	---

<code><timeout></code>	Wait timeout in milliseconds. If timeout is 0, then the function does not block. If timeout is -1, then the function will wait indefinitely for at least one file to exist that matches the file pattern. Any other negative value is illegal. On a computer where millisecond timing accuracy is not available, timeout is rounded up to the nearest legal value available on that system.
------------------------------	---

<code><poll_interval></code>	Polling interval in milliseconds to look for the existence of file(s). On a computer where millisecond timing accuracy is not available, the polling interval is rounded up to the nearest legal value available on that system. If the poll interval exceeds the timeout value then, it will be rounded up to timeout value.
<code><max_match ></code>	Optional. Specifies the maximum number of matched file names that function should return. The default value is 0. -1 specifies all the matched file names.
<code>< file_name_list ></code>	Optional. Output varchar variable that returns the list of matched file names. Order of the file names in the list is determined the way operating system returns the file names.
<code>< list_size></code>	Optional. Output integer variable that returns the list size.
<code><list_separator></code>	Optional. File name list separator character(s). Default value is comma (,).

Details:

This function waits a maximum of up to timeout interval for at least one file to exist that matches the pattern. Poll interval determines how often to poll for files.

❖ Example

This function is used in a script at the beginning of a job. A job will suspend until a file is present, as shown in the following business use case example:

During the night, an external process puts source files in a file system that the system can access. Usually this process is finished at 1:00 AM, but it can be later. You schedule the job to start at 1:00 AM, but in the first step of the job use a script that checks for the existence of the last file. If the file does not exist, the job will wait for some time and try again later. Once the file is present, the job will continue. A timeout needs to be set to stop the job when the file is still not present at 9:00 AM in the morning.

9.3.162 week_in_month

Determines the week in the month in which the given date falls.

This function considers the first week of the month to be first seven days. The day of the week is ignored when calculating the weeks.

≡ Syntax

```
week_in_month(<date1>)
```

Return value

int

The number from 1 to 5 that represents which week in the month that `<date1>` occurs.

Where

`<date1>` The source date.

❖ Example

Function	Results
<code>week_in_month(to_date('Jan 22, 1997', 'mon dd, yyyy'))</code>	4
<code>week_in_month(to_date('Jan 21, 1997', 'mon dd, yyyy'))</code>	3

9.3.163 week_in_year

Returns the week in the year in which the given date falls.

This function returns the week in the year in two ways:

- 'WW' - Absolute week number of the given date.
- 'IW' - ISO week number of the given date.

≡ Syntax

```
week_in_year(<inputdate>, '<weektype>')
```

Return value

int

Where

`<inputdate>` The source date

`<weektype>` Optional. The values are 'WW', or 'IW'. Default value is "WW".

Description

The number from 1 to 53 that represents the week number in a year. This function considers the first week of the year to be the first seven days while determining the absolute week number. Under the ISO standard, a week always begins on a Monday, and ends on a Sunday. The first week of a year is that week which contains the first Thursday of the year. An ISO week number may be between 1 and 53. Under the ISO standard, week 1 will always have at least 4 days. If 1-Jan falls on a Friday, Saturday, or Sunday, the first few days of the year are defined as being in the last (52nd or 53rd) week of the previous year.

❖ Example

Some business applications use week numbers to categorize dates. For example, a business may report sales amounts by week, and identify each period as "9912", representing the 12th week of 1999. (An ISO week is more meaningful than an absolute week for such a purpose.)

Following are more example results for `week_in_year` applied to three different input dates:

Function	Results
<code>week_in_year(to_date('Jan 01, 2001','mon dd, yyyy'))</code>	1
<code>week_in_year(to_date('2005.01.01','yyyy.mm.dd'),'WW')</code>	1
<code>week_in_year(to_date('2005.01.01','yyyy.mm.dd'),'IW')</code>	53

9.3.164 WL_GetKeyValue

Returns the value of a given keyword in Web log search strings.

☞ Syntax

```
WL_GetKeyValue(<string>, <keyword>)
```

❖ Example

If you search for **BusinessObjects** on Google, the following appears in a Web log:

```
GET "http://www.google.com/search?hl=en&lr=&safe=off&q=BusinessObjects&btnG=Google+Search"
WL_GetKeyValue('http://www.google.com/search?hl=en&lr=&safe=off&q=BusinessObjects&btnG=Google+Search','q') returns
'BusinessObjects'.
```


9.3.165 word

Returns one word out of a string.

Syntax

```
word(<input_string>, <word_num>)
```

Return value

varchar

A string containing the indicated word. The return type is the same as `<input_string>`.

Where

<code><input_string></code>	The source string.
<code><word_num></code>	A nonnegative integer specifying the index of the target word in the string. The first word in a string is word number 1. If <code><word_num></code> is 0 or greater than the number of words in <code><input_string></code> , then the word function returns a NULL string.

A word is defined to be any string of consecutive non-white space characters terminated by white space, or the beginning and end of `<input_string>`. White space characters are the following:

- Space
- Horizontal or vertical tab
- Newline
- Linefeed

Example

Function	Results
<code>word('Accounting Department', 1)</code>	'Accounting'
<code>word('Accounting', 1)</code>	'Accounting'
<code>word('Accounting', 2)</code>	NULL

9.3.166 word_ext

Returns the word identified by its position in a delimited string.

This function is useful for parsing Web log URLs or file names.

≡ Syntax

```
word_ext('<string>', <word_num>, '<separator(s)>')
```

Return value

varchar

A string containing the indicated word. Return type is the same as `<string>`.

Where

<code><string></code>	The source string.
<code><word_num></code>	A nonnegative integer specifying the index of the target word in the string. The first word in a string is word number 1. If <code><word_num></code> is 0 or greater than the number of words in <code><string></code> , then the word function returns a NULL string.
<code><separator(s)></code>	Any character specified.

A word is defined to be any string of consecutive non-white space characters terminated by white space, or the beginning and end of `<string>`. White space characters are the following:

- Space
- Horizontal or vertical tab
- Newline
- Linefeed

⚙ Example

Function	Results
<code>word_ext('www.sap.com', 2, '.')</code>	'sap'
<code>word_ext('www.cs.wisc.edu', -2, '.')</code>	'wisc'
	A negative word number means count from right to left.

Function	Results
<code>word_ext('www.cs.wisc.edu', 5, '.')</code>	NULL
<code>word_ext('aaa+=bbb+=ccc+zz=dd', 4, '+=')</code>	'zz' If 2 separators are specified (+=), the function looks for either one.
<code>word_ext(',,,,,aaa,,,,bb,,,c ', 2, ',')</code>	'bb' This function skips consecutive delimiters.

9.3.167 workflow_name

Returns the name of the current work flow.

Syntax

```
workflow_name()
```

Return Value

varchar

Details

In cases where several work flows enclose this function, the function returns the name of the inner most work flow. If no work flow is found, the function returns the job name.

Example

```
print('Work Flow Name: [workflow_name()]')
```

9.3.168 year

Determines the year in which the given date falls.

Syntax

```
year(<date1>)
```

Return value

int

The number that represents the year component of <date1>.

Where

<date1>

The source date.

Example

Function	Results
<code>year(to_date('Jan 22, 1997','mon dd, yyyy'))</code>	1997
<code>year(to_date('03/97','mm/yy'))</code>	1997
<code>year(to_date('03/19','mm/yy'))</code>	2019

9.4 Custom functions

You can create your own functions by writing script functions in SAP Data Services scripting language using the smart editor. Saved custom functions appear in the function wizard and the smart editor under the [Custom](#) category. They also are displayed in the object library under the [Custom Functions](#) tab. You can edit and delete custom functions from the object library.

In the object library on the [Custom Functions](#) tab, there are multiple categories of custom functions:

- Custom Functions
- Validation Functions:
 - Imported from Information Steward: These functions were created in Information Steward and imported; they are not editable in Data Services.

- Locally Created: These are reusable, custom, validation functions created in Data Services.

Consider these guidelines when you create your own functions:

- Functions can call other functions.
- Functions cannot call themselves.
- Functions cannot participate in a cycle of recursive calls. For example, function A cannot call function B, which calls function A.
- Functions return a value.
- Functions can have parameters for input, output, or both. However, data flows cannot pass parameters of type output or input/output.

Before creating a custom function, you must know the input, output, and return values and their data types. The return value is predefined to be `Return`.

Related Information

[Scripting Language \[page 1259\]](#)

9.4.1 Creating a custom function

1. Choose **Tools** > **Custom Functions**.
2. In the Custom Function list, right-click and select **New**.

Alternatively, from the object library, right-click and select **New** in the Custom Functions tab.

3. Enter the name of the new function.
4. Enter a description for your function.
5. Click **Next** to open the smart editor.

In the smart editor, you can define the return type, parameter list, and any local variables to be used in the function.

6. In the Variables tab, right-click **Return** and select **Properties...**

By default, the return data type is set to `int`. To change this, select another return data type from the Data type list. Click **OK**.

7. In the Variables tab, right-click **Parameters** and choose **Insert**.
8. Define parameter properties by choosing a Data type and a Parameter type (Input, Output, or Input/Output).

i Note

Data flows cannot pass variable parameters of type output and input/output.

9. Click **OK**.

Repeat steps 7 - 9 for each parameter required in your function. After you add one parameter, the right-click menu allows you to choose where to insert each new one.

Use this menu to create, delete, or edit variables and parameters.

10. To define variables used by the function, but not passed outside the function, right-click *Local* and choose *Insert*.

11. Choose a data type in the Variable Properties window and click *OK*.

Repeat this step for each variable required in your function.

12. Complete the text for your function.
13. Click the *Validate* icon to validate your function.



If your function contains syntax errors, you will see a listing of those errors in an embedded pane below the editor.

14. To see where the error occurs in the text, double-click on an error.

The smart editor redraws to show the location of the error.

15. When your function is valid, click *OK* to save the function, and the variables and parameters inside.

Variables and parameters for an existing custom function are local to each function. Therefore, they are not displayed in the Variables and Parameters window (accessible from ► *Tools* ► *Variables* ►). Variables and parameters for custom functions can be viewed in the smart editor library, under the Variables tab, when you edit the custom function.

Related Information

[Smart editor \[page 299\]](#)

9.4.2 Editing an existing function

Choose ► *Tools* ► *Custom Functions* ►, select the function and select *Next*.

The smart editor opens with the function displayed and the variables and parameters that exist for this function shown under the Variables tab.

Alternatively:

- Go to the Custom Function tab in the object library and double-click the function you want to edit.
- Go to the Custom Function tab in the object library, right-click the function, and select *Edit*.

Related Information

[Custom functions \[page 1242\]](#)

9.4.3 Replicating a custom function

1. From the object library, right-click a custom function and select *Replicate*.

The Custom Function editor opens with only the Function name field enabled.

2. Enter a new name for the function.

The name must be valid and different from the original name.

3. Click *Finish*.

The new custom function appears in the object library.

9.4.4 Deleting a custom function

Go to the Custom Function tab in the object library, right-click the function, and select *Delete*.

If you delete a function from the list of custom functions, you must remove references to the function from expressions in scripts, conditionals, queries, and other custom functions.

9.5 About procedures

Data Services supports the use of stored procedures for DB2, ODBC, Oracle, Microsoft SQL Server, SAP HANA, SAP Sybase SQL Anywhere, SAP ASE, SAP Sybase IQ, and Teradata databases. You can call stored procedures from the jobs you create and run in Data Services.

9.5.1 Overview

A stored procedure is a generic term used to describe an executable object, or a named entity that is stored in a database and can be invoked using input and output parameters. Generally, a stored procedure is one or more precompiled SQL statements. By calling a stored procedure from within Data Services, you can invoke business logic you have already coded thus enabling you to quickly and conveniently develop data extraction and data management tasks. Stored procedures can also be used to:

- Maintain business logic rules and provide a single point of control to ensure rules are accurate and enforced
- Significantly reduce network overhead with client/server applications because:
 - Procedures are stored on the database server
 - Compiled execution plans for procedures are retained in the data dictionary

Data Services supports stored procedures for DB2, ODBC, Oracle, Microsoft SQL Server, SAP HANA, SAP Sybase SQL Anywhere, SAP ASE, SAP Sybase IQ, and Teradata databases. Data Services also supports stored functions and packages for Oracle databases. Queries, scripts, conditionals, and custom functions can all be configured to include stored procedures, stored functions, and packages.

Stored procedures must exist in the database before you can use the procedure in Data Services. Create a stored procedure in a database using the client tools provided with the database, such as Oracle SQL *Plus. After it is created, a stored procedure can be called by users who have execution privileges for the procedure. After they are imported into Data Services, stored procedures can be used like functions in Data Services jobs.

Stored procedures include parameters. Each parameter has a name, a data type, and a mode (`IN`, `INOUT`, or `OUT`). A stored procedure can use a `NULL` or default parameter value for its input and can produce more than one output parameter value.

9.5.2 Requirements

To use stored procedures with Data Services, the following requirements must be met:

- The client and server versions must match.
- Only user-defined stored procedures can be used. Data Services does not support stored procedures provided by a database system.
- User-defined stored procedures or stored functions must meet the following additional requirements:
 - The return type (the data type of the result value) must be a Data Services supported data type, such as `varchar`, `date`, or `int`.

i Note

This release does not support the long data type in stored procedures.

- The name of the stored procedure—the combination of the datastore name, owner name, and procedure name—must be unique. Data Services only imports the first procedure or function with a particular name.
Using Oracle for example, if you have multiple procedures or functions with the same name in a package and want to use all of them, you must rename the procedures, giving each a unique name. A procedure is overloaded when multiple versions exist in a particular package. Overloading is not supported in Data Services.
- Data Services validates the user name specified in the datastore which was used to import the stored procedure. This ensures that the data flow that calls a stored procedure enforces restrictions associated with that procedure, such as:
 - Data input restrictions
 - Execution privileges
- To use stored procedures with Teradata, you must first enable the use of stored procedures within the Teradata ODBC driver options and invoke the stored procedures in the same session mode in which it was enabled.
For Windows, in the [Teradata ODBC Driver Options](#) screen used for configuration of the Teradata ODBC driver, check the box [Disable CALL to EXEC Conversion](#), set the session mode as desired in the [Session Mode](#) drop-down list, and click [OK](#) to save the change. You must invoke the stored procedures in the same session mode.

For UNIX platforms, edit the file `odbc.ini` to set the option **Disable CALL To EXEC Conversion=YES**. Set the session mode option as desired, and save the file. You must invoke the stored procedures in the same session mode.

For more information about Teradata ODBC driver configuration, see your Teradata ODBC driver configuration documentation.

9.5.3 Creating stored procedures in a database

This section provides:

- An example scenario for using a stored procedure
- Tips for creating stored procedures on each database
- Example syntax for DB2, Oracle, SAP HANA, SQL Server, and SAP ASE databases based on the example scenario

Since syntax varies between databases, the SQL statements needed to create a stored procedure for any scenario vary. The client tools used to pass the SQL statements to the database server also vary. Refer to your database documentation for more detailed descriptions and examples regarding how to create stored procedures.

In the following example, the source database has a stored procedure `Get_emp_rec` that retrieves an employee's name and hire date from the `Employee` table using a given employee number. The stored procedure takes an input parameter `Emp_number` and returns two values `Emp_name` and `Hire_date` via output parameters.

The schema of the database table `Employee` is as follows:

Name	Null?	Type
empno	Not Null	Integer
ename	Null	Varchar(20)
job	Null	Varchar(9)
mgr	Null	Integer
hiredate	Null	Date
sal	Null	Decimal(7,2)
comm	Null	Decimal(7,2)
deptno	Null	Integer

9.5.3.1 Creating stored procedures in DB2

In DB2, a parameter can be `IN`, `OUT` or `INOUT`. Any SQL data type can be used as a data type of a parameter. User defined data types are not supported.

❖ Example

The DB2 syntax for the stored procedure

```
CREATE PROCEDURE GET_NAME_USING_ID (IN NID INTEGER, OUT outVar varchar(20))
language SQL
reads sql data
BEGIN
select first_name into outVar FROM CONTACT where id = NID;
END
```

`Reads sql data` and `language SQL` are two of the many options that DB2 stored procedures support. Refer to the DB2 documentation for the DB2 stored procedure options and their meaning.

Related Information

[Creating stored procedures in a database \[page 1247\]](#)

9.5.3.2 Creating stored procedures in Oracle

Oracle supports both stored procedures and stored functions. Any stored procedure that returns a value is called a stored function. Oracle is the only database to allow return values with data types other than an integer.

In Oracle, stored procedures are created using the `CREATE [OR REPLACE] PROCEDURE` statement, and stored functions are created using the `CREATE [OR REPLACE] FUNCTION` statement. The `OR REPLACE` option allows you to override an existing definition of a procedure or function with the same name.

Oracle supports `IN`, `OUT`, and `INOUT` parameters. An `INOUT` parameter allows you to pass in a parameter, modify it, and return the modified value. You can use the `DEFAULT` keyword or the assignment operator to give an `IN` parameter a default value. When an `IN` parameter has a default value, you can omit the parameter from the argument list when you call the procedure. If you do specify an argument value in the call, the specified value overrides the default value. `INOUT` and `OUT` parameters must be specified.

❖ Example

Syntax

The Oracle syntax for the stored procedure:

```
CREATE OR REPLACE PROCEDURE
get_emp_rec (Emp_Number IN NUMBER,
Emp_Name OUT VARCHAR, Emp_Hiredate OUT DATE) AS
BEGIN
SELECT ename, hiredate
```

```

INTO Emp_Name, Emp_Hiredate
FROM Employee
WHERE empno = Emp_Number;
END;
/

```

In this example, the parameters are declared as IN and OUT, but Oracle also supports an INOUT parameter type, which allows you to pass in a parameter, modify it, and return the modified value. INOUT parameters can take default values. This means that the parameter can be omitted from the actual parameter list when you call the procedure. INOUT and OUT parameters must be specified.

An Oracle package is an encapsulated collection of related program objects (e.g., procedures, functions, variables, constants, cursors, and exceptions) stored together in the database. Data Services allows you to import procedures or functions created within packages and use them the same way as top-level procedures or functions.

Data Services does not support overloading of procedures or functions. Overloading a procedure means creating multiple procedures with the same name in the same package, each taking arguments of a different number or data type. If you have multiple procedures or functions with the same name in the same package (and you want to use all of them), you need to rename the procedures or functions so that they all have distinct names. Otherwise, Data Services will only import the first procedure or function by that name.

Related Information

[Creating stored procedures in a database \[page 1247\]](#)

9.5.3.3 Creating stored procedures in SAP HANA

SAP Data Services supports SAP HANA stored procedures with zero, one, or more output parameters.

Data Services supports scalar data types for input and output parameters. Data Services does not support table data types. If you try to import a procedure with table data type, the software issues an error. Data Services does not support data types such as binary, blob, clob, nclob, or varbinary for SAP HANA procedure parameters.

Procedures can be called from a script or from a Query transform as a new function call.

❖ Example

Syntax

The SAP HANA syntax for the stored procedure:

```

CREATE PROCEDURE GET_EMP_REC (IN EMP_NUMBER INTEGER, OUT EMP_NAME
VARCHAR(20), OUT EMP_HIREDATE DATE) AS
BEGIN
    SELECT ENAME, HIREDATE
        INTO EMP_NAME, EMP_HIREDATE
    FROM EMPLOYEE
    WHERE EMPNO = EMP_NUMBER;

```

```
END;
```

Limitations

SAP HANA provides limited support of user-defined functions that can return one or several scalar values. These user-defined functions are usually written in L. If you use user-defined functions, limit them to the projection list and the GROUP BY clause of an aggregation query on top of an OLAP cube or a column table. These functions are not supported by Data Services.

SAP HANA procedures cannot be called from a WHERE clause.

Related Information

[Creating stored procedures in a database \[page 1247\]](#)

9.5.3.4 Creating stored procedures in MS SQL Server or SAP ASE

In Microsoft SQL Server and SAP ASE, a stored procedure may have a group number (between 1 and 32767), which is an optional integer used to group procedures of the same name so they can be dropped together with a single `DROP PROCEDURE` statement.

The first procedure you create with a name automatically is assigned group number 1. While you can execute this procedure without specifying the group number, other procedures with the same procedure name must be called with the group number. For example, `execute orderproc;2` and so on. Data Services imports the procedure with the group number 1, i.e., the first procedure, unless you specify the name including the group number, such as `orderproc;2`.

A parameter can be either an input parameter (default) or an output parameter. A parameter can not be both input and output. In the following sample, the first parameter `@Emp_Number` is an input parameter and the other two parameters, `@Emp_Name` and `@Emp_Hiredate`, are output parameters. Output parameters are specified using the keyword `OUTPUT`. Input parameters can take default values (database defaults). If a default value is defined, the procedure can be executed without specifying a value for that parameter. You specify a parameter name with "@" as the first character.

An optional integer return value is automatically added to each stored procedure to store its return status. In the following example, if the `SELECT` statement is successful, "0" is returned as the return value. The parameter name of the return value is `RETURN_VALUE`. Microsoft SQL Server and SAP ASE stored procedures can be used like Oracle stored functions in Data Services because they include this return value.

A user-defined function is a new feature of SQL Server 2000. Data Services supports user-defined stored procedures, but not user-defined functions at this time.

❖ Example

The MS SQL Server and SAP ASE syntax

```
CREATE PROCEDURE GET_EMP_REC @Emp_Number int, @Emp_Name varchar(20) OUTPUT,  
@Emp_Hiredate datetime OUTPUT AS  
SELECT @Emp_Name = ename, @Emp_Hiredate = hiredate  
FROM Employee  
WHERE empno = @Emp_Number  
RETURN 0
```

Related Information

[Creating stored procedures in a database \[page 1247\]](#)

9.5.4 Importing metadata for stored procedures

Use the Import by Name command to import stored procedures into Data Services. You must know the name of the stored procedure, stored function, or package you want to import. Use the name of the stored procedure, the package name only, or, if you are using a Microsoft SQL Server or SAP ASE database, use the name of the stored procedure and the group number, such as `orderproc;2`.

Stored procedures are created and structured very much like functions. Imported stored procedures appear on the Datastores tab of the object library. Each is listed in the Functions category of the database datastore from which the procedure was imported. It is displayed with the following syntax:

```
<datastore>.<owner>.<proc_name>.
```

If a stored procedure was defined within an Oracle package, the name of the procedure is displayed as

```
<datastore>.<owner>.<package>.<proc_name>.
```

To view the signature of the procedure, right-click the imported stored procedure and select Open (or simply double-click the procedure).

- A stored procedure has a signature that consists of its defined parameters including two output parameters added by Data Services during import, `AL_SP_RETCODE` and `AL_SP_ERRMSG`.
- If the database you are using provides a return code for a stored procedure, it is displayed in the signature as `AL_SP_RETURN`.
- Each parameter has a name, a data type, and a mode (`IN`, `INOUT`, or `OUT`). The mode is indicated by icons shown before the parameter name.
- Oracle stored functions have a signature that consists of parameters and a return data type. When you call a function, the return value is named `AL_SP_RETURN`.

i Note

If you change the signature of a stored procedure, for example you add a new parameter, re-import the metadata for the procedure from the database.

Related Information

[Checking execution status \[page 1257\]](#)

9.5.5 Structure of a stored procedure

Imported information for a stored procedure includes:

- Name and owner
- Return data type if the stored procedure has a return value or code
- Parameters

For each parameter, Data Services imports the following information:

- Name
- Mode (IN, OUT, or INOUT)
- data type (length, precision and scale if applicable)

i Note

In an Oracle stored procedure declaration, it is illegal to constrain char and varchar2 parameters with a length, or to constrain number parameters with a precision and/or scale. These constraints are taken from the procedure arguments. Therefore, the length of a char or varchar and the precision and scale of a number data type parameter are not stored in the data dictionary.

However, Data Services does not allow the length of a varchar data type or the precision of a number data type to be 0. (Data Services converts char data types in Oracle to varchar.) When importing a stored procedure, Data Services fills in the default length for a varchar or the precision and scale for a number parameter. For varchar, Data Services sets the length to 4000. For number, Data Services sets precision and scale to the values set in the Advanced section of the Create New Datastore window. At runtime, the length of varchar and the precision and scale of number come from the procedure arguments.

- Position within the parameter list
- Nullable (whether a default value exists)

Data Services omits a parameter if any of the following data type conditions is true:

- Is not a Data Services supported data type or is `long`
- Data type is missing in the database dictionary
- Is a composite data type (e.g., table, record, or cursor)

After Data Services imports a stored procedure, the Designer reports warnings about any omitted parameters.

i Note

Even if a stored procedure has an omitted parameter, the procedure is callable if the parameter has a default value defined. If the omitted parameter does not have a default value, the procedure is not callable and Data Services throws a runtime error.

If a stored procedure has a return value, its result type (the data type of the result value) must be a Data Services supported database server data type. Otherwise, the procedure cannot be imported and used within Data Services applications.

For each stored procedure, Data Services automatically adds two optional output parameters, `AL_SP_RETCODE` and `AL_SP_ERRMSG`. These parameters allow you to check the execution status of the stored procedure.

Related Information

[Checking execution status \[page 1257\]](#)

[Data Types \[page 305\]](#)

9.5.6 Calling stored procedures

You can call a stored procedure just as you call custom functions in Data Services.

- When you call a stored procedure inside a script, conditional or custom function, Data Services evaluates the call as an independent SQL statement.
- When you call a stored procedure in a query's column mapping or WHERE clause, (Oracle only) Data Services can combine the SQL for the stored procedure with the SQL for the query.
- When you call a stored procedure in a SQL transform, you must use proper syntax. Data Services does not validate SQL in a SQL transform.
- A stored procedure can be used within an expression. For example:

```
ora_ds.acta_user.get_emp_sal (EMPLOYEE.NAME) + 10  
ora_ds.acta_user.get_emp_sal (EMPLOYEE.NAME) < 100
```

Related Information

[About procedures \[page 1245\]](#)

[Stored procedure input parameters \[page 1253\]](#)

[Calling stored procedures from scripts \[page 1254\]](#)

[Calling stored procedures from queries \[page 1255\]](#)

[Without the function wizard \[page 1256\]](#)

9.5.6.1 Stored procedure input parameters

Stored procedures can require input values for some parameters but not others. You must determine the requirements of the stored procedure and prepare the appropriate inputs. In Data Services, input parameters can be constants or expressions. In addition, Data Services supports the following:

- If an input parameter has a default value, you can omit the parameter in the call by leaving the parameter unspecified in the function wizard. The database will use the parameter's default value when evaluating the stored procedure call.

- All `OUT` or `INOUT` parameters must be specified except the two Data Services adds at import: `AL_SP_RETCODE` and `AL_SP_ERRMSG`.
- All `OUT` or `INOUT` parameters must be variables (except in a query output schema). A variable must be declared before use.
- If a stored procedure returns a value (such as an Oracle function or a MS SQL Server or SAP ASE stored procedure), you can use the procedure in an expression.
- You can call a stored procedure as a step inside an output schema of a query, script, or custom function. For example:

```
As an Oracle user you create two Oracle stored procedures: GET_EMP_SALARY and
SET_EMP_SALARY
CREATE OR REPLACE FUNCTION GET_EMP_SALARY(Emp_Name IN VARCHAR)
RETURN NUMBER AS
Return_value NUMBER;
BEGIN
SELECT sal
INTO Return_value
FROM Employee
WHERE ename = Emp_Name;
RETURN Return_value;
END;
CREATE OR REPLACE PROCEDURE SET_EMP_SALARY(Emp_Name IN VARCHAR,Emp_Sal IN
NUMBER) AS
BEGIN
UPDATE Employee
SET sal = Emp_Sal
WHERE ename = Emp_Name;
END;
```

Related Information

[About procedures \[page 1245\]](#)

[Calling stored procedures \[page 1253\]](#)

9.5.6.2 Calling stored procedures from scripts

After importing the stored procedures into an Oracle datastore, you can use the stored procedures in a script. For example:

```
$new_sal = ora_ds.bodi_user.get_emp_salary('MARTIN') + 100;
ora_ds.bodi_user.set_emp_salary('MARTIN', $new_sal);
```

Related Information

[About procedures \[page 1245\]](#)

[Calling stored procedures \[page 1253\]](#)

9.5.6.3 Calling stored procedures from queries

You can call stored procedures from Query transforms in the mapping and WHERE clauses, or in a Query transform output schema.

Related Information

[About procedures \[page 1245\]](#)

[Calling stored procedures \[page 1253\]](#)

[Importing metadata for stored procedures \[page 1251\]](#)

9.5.6.3.1 Mapping and WHERE clauses

In a query's column mapping or WHERE clause, you can use a stored procedure in an expression, subject to the same constraints as other expressions. However, when stored procedures are used in a query's column mapping or WHERE clause, SAP Data Services can combine the SQL statement for the stored procedure call with the SQL statement for the query and submit one request to the database.

For the software to combine the SQL statements, the stored procedure must meet the following conditions:

- It must be from an Oracle database.
- It must be a stored function.
- All its parameters must be IN parameters; none can be an OUT or INOUT parameter.
- All parameter values must be specified. You cannot leave a parameter unspecified, such as to use the default value.
- All parameter values must be constants or expressions with only constants or database table column names; none can have functions or operations.
- The data types of the procedure's parameters and the return type (the data type of its result value) must be supported data types. The software must not omit any of the parameters when importing the procedure.
- Data Definition Language (DDL) operations like creating tables, views etc. and transaction control statements such as `COMMIT`/`ROLLBACK` cannot be performed inside the stored procedure.

In addition, when importing the function, you must select the Callable from SQL expression check box on the Import By Name window.

The software will push down a stored procedure call to Oracle if the first five conditions are met and the [Callable from SQL expression](#) check box is selected. If a stored procedure call contains any DDL statements, it ends the current transaction with `COMMIT` or `ROLLBACK`, or it issues any `ALTER SESSION` or `ALTER SYSTEM` commands. Therefore, you should deselect the [Callable from SQL expression](#) option on the Import By Name window to ensure that the function call will not be pushed down to database.

When a stored procedure call cannot be pushed down to database, it is submitted to database as a separate SQL statement.

Related Information

[About procedures \[page 1245\]](#)

[Calling stored procedures \[page 1253\]](#)

9.5.6.3.2 Query and transform output schema

When a stored procedure can provide at least one return value or has any `INOUT` or `OUT` parameters other than `AL_SP_RETCODE` or `AL_SP_ERRMSG`, you can call the procedure by including it as an object in the output schema of a query.

When using a stored procedure in a query output schema, you must map `INOUT` or `OUT` parameters and return values to a column of the output schema. In this case, you cannot map these parameters to variables.

Related Information

[About procedures \[page 1245\]](#)

[Calling stored procedures \[page 1253\]](#)

9.5.6.4 Without the function wizard

You can enter an existing stored procedure call on the [Mapping](#) tab or the [Where](#) tab in a query editor.

When entering a stored procedure call without using the function wizard:

- Match the parameter values you enter to the signature of the stored procedure in Data Services. Values must be specified in the same order that the parameters are defined.
- Use `AL_UNSPECIFIED_PARAM` in place of a missing parameter, such as when you want to use the default value for an `IN` parameter.
- Use the correct naming convention for the stored procedure:
 - `<datastore>.<owner>.<proc_name>`
 - `<datastore>.<owner>.<package>.<proc_name>`

If the name is case-sensitive in the database (and not all uppercase), enter the name as it appears in the database and use double quotation marks (") around the name to preserve the case.

The two output parameters added by Data Services during import, `AL_SP_RETCODE` and `AL_SP_ERRMSG`, are optional. You do not need to provide arguments for these two parameters if you are not interested in the values stored in them.

Related Information

[About procedures \[page 1245\]](#)

[Calling stored procedures \[page 1253\]](#)

[Checking execution status \[page 1257\]](#)

9.5.6.4.1 Including a stored procedure in a query output schema

1. In a data flow diagram, click a query name to open the query editor.
2. In the query editor, right-click the query name in the output schema and select [New Function Call](#).

Alternately, you can right-click an existing stored procedure object in the output schema area of the query editor and select [Modify Function Call](#).

SAP Data Services provides a function wizard to help you include input and output parameters and the return value for a procedure call. The imported stored procedures are listed for each datastore on the first page of the function wizard.

3. Select a function category. Notice that the function names change depending on the category you select.
4. Under Function name, select a stored procedure, then click [Next](#).
5. In the [Define Input Parameter\(s\)](#) window, enter the input parameters for the stored procedure, then click [Next](#).
6. In the [Select Output Parameter](#) window, select the procedure's output parameters that you want to map to the query output.

In the [All output parameters](#) box, select the parameters you want to map, and click the arrow key to move them into the [Selected output parameters](#) box.

You can map more than one INOUT or OUT parameter and the return value as output columns from a single procedure call.

7. Click [Finish](#).

The software adds those parameters to the query's output schema. In this example, the output schema of the Query transform would have two columns, EMP_NAME and EMP_HIREDATE.

9.5.7 Checking execution status

To allow you to monitor the execution status of a stored procedure, Data Services adds two parameters to the signature of each stored procedure when importing the procedure:

- AL_SP_RETCODE, varchar(256). There are three possible values:
 - ACTA_SP_OK — The stored procedure call succeeded.
 - ACTA_SP_DB_CONN_EXCEPTION — The database connection cannot be created because of a connection error, invalid user, password, or host name.

- `ACTA_SP_CALL_ERROR` — The connection completes but the call fails in the database. Details for the cause of the error are available in `AL_SP_ERRMSG`.
- `AL_SP_ERRMSG`, `varchar(1024)`

Data Services adds these two parameters to the end of the signature, following any user-defined parameters.

With these parameters, there are two techniques you can use to handle errors if a stored procedure fails during execution:

- Throw an error that stops the job immediately. An error message is logged in the `errorlog.txt` file. To use this method, do not map as output parameters the two parameters Data Services adds to imported stored procedures.
- Save the Oracle error messages into the `AL_SP_ERRMSG` parameter. This method allows the query, script, conditional, or custom function that contains the stored procedure call to continue its execution. To use this method, define parameters for `AL_SP_RETCODE` and `AL_SP_ERRMSG`. Use the execution status stored in `AL_SP_RETCODE` to control your application logic. For example, if you store the value of `AL_SP_ERRMSG` in a variable, you can use the print function to print the variable to the job log. With this method, the error message is not logged in the `errorlog.txt` file.

10 Scripting Language

You can use the SAP Data Services scripting language to write scripts and custom functions (also known as user-script functions). Further, you can use scripting language to write expressions such as complex column mapping expressions and WHERE clause conditions.

The software uses this language internally to save the objects you create into repository tables. You can export saved objects to a file (.atl) and later import the file to another repository. This technique is commonly used when migrating to a new environment or when backing up a repository before initiating an upgrade.

10.1 Using the scripting language

1. Open an editor for an object in the Designer workspace such as a script, conditional, or transform.
2. Inside any of these editors, use the smart editor to enter a script or expression using the language syntax described in this section.

Related Information

[Language syntax \[page 1259\]](#)

[Sample scripts \[page 1270\]](#)

10.2 Language syntax

The syntax used in the scripting language can be used in an expression as well as in a script. With the scripting language, you can assign values to variables, call functions, and use standard string and mathematical operators.

10.2.1 Syntax for statements in scripts

Statements in scripts are detailed steps written in the flow of logic.

Statements in a script object or custom function must end with a semicolon (;). Comment lines must start with a (#) character.

Related Information

[Sample scripts \[page 1270\]](#)

10.2.2 Syntax for column and table references in expressions

The Data Services Scripting Language recognizes column and table names without special syntax

Expressions are a combination of constants, operators, functions, and variables that evaluate to a value of a given data type. Expressions can be used inside script statements or added to data flow objects. Because expressions can be used inside data flow objects, they often contain column names.

No special syntax is required for column or table names. For example, you can indicate the `start_date` column as the input to a function as follows:

```
to_char(start_date, 'dd.mm.yyyy')
```

The column `start_date` must be in the input schema of the query.

If there is more than one column with the same name in the input schema of a query, indicate which column is included in an expression by qualifying the column name with the table name. For example, indicate the column `start_date` in the table `status` as follows:

```
status.start_date
```

Note that column and table names as part of SQL strings may require special syntax based on the DBMS that the SQL is evaluated by. For example, select all rows from the `LAST_NAME` column of the `CUSTOMER` table as follows:

```
sql('oracle_ds', 'SELECT CUSTOMER.LAST_NAME  
FROM CUSTOMER')
```

Because the column name is all upper case in Oracle, no special syntax is required. However, select all rows from the `start_date` column in the Oracle table `Status` as follows:

```
sql('oracle_ds', 'SELECT "Status"."start_date"  
FROM "Status"')
```

The table name and column are defined in Oracle to have a special case. To preserve this case, enclose the identifiers in double quotation marks.

10.2.3 Strings

This section discusses three situations about using strings syntax.

- Quotation marks
- Escape characters
- Trailing blanks

Related Information

[Quotation marks \[page 1261\]](#)

[Escape characters \[page 1261\]](#)

[Trailing blanks \[page 1262\]](#)

[NULL values and empty strings \[page 1265\]](#)

10.2.3.1 Quotation marks

The type of quotation marks to use in strings depends on whether you are using identifiers or constants.

Identifier	Name of an object (for example, table, column, data flow, or function).
Constant	<p>A fixed value used in computation. There are two types of constants:</p> <ul style="list-style-type: none">• String constants (for example, 'Hello World' or '1995.01.23')• Numeric constants (for example, 2.14)

Identifiers need quotation marks if they contain special (non-alphanumeric) characters. For example, you need a double quote for `" compute large numbers "` because it contains blanks.

Use single quotes for string constants.

10.2.3.2 Escape characters

If a constant contains a single quote (') or backslash (\) or another special character, then those characters need to be escaped.

For example, the following characters must be preceded with an escape character to be evaluated properly in a string. Data Services uses the backslash (\) as the escape character.

Character	Example
single quote (')	'World\'s Books'
backslash (\)	'C:\\temp'

10.2.3.3 Trailing blanks

Data Services does not strip trailing blanks from strings that are used in scripts or custom functions. To remove trailing blanks, use the `rtrim` or `rtrim_blank` function.

10.2.4 Variables

Variable names must be preceded by a dollar sign (\$).

- Local variables used in a script or expression must be defined in the job or work flow that calls the script using the [Variables and Parameters](#) window.
- Global variables used in a script or expression must be defined at the job level using the [Variables and Parameters](#) window.
- The return value must be passed outside the function using the following statement:

```
RETURN (<expression>)
```

where expression defines the value to be returned.

- Local variables used in a custom function must be defined using the smart editor.
- Existing variables and parameters displayed in the smart editor are filtered by the context from which the smart editor is opened.

Related Information

[Creating a custom function \[page 1243\]](#)

10.2.5 Variable interpolation

You can embed expressions within constant strings, and Data Services will evaluate the variables and substitute the value into the string—the concatenation operator (`||`) is not required.

Here is an example using the concatenation operator:

```
$st_date=sql('warehouse_ds','SELECT max(start_timestamp)
FROM dw_process_log
WHERE table_name='\''||$Table_Name||'\''')
print('The value of the start date
is:'||sysdate()+5);
```

The statement can be simplified to the following:

```
$st_date=sql('warehouse_ds','SELECT max(start_timestamp)
FROM dw_process_log
WHERE table_name={$Table_Name}')
print('The value of the start date
is:[sysdate()+5]');
```


Curly braces ({}) and square brackets ([]) enclose the embedded expressions:

- The square brackets ([]) indicate that the value of the expression should be substituted.
- The curly braces ({}) indicate that the value of the expression should be quoted with single quotation marks.

Strings including curly braces or square brackets cause a processing error. You can avoid the error by preceding the braces or brackets with a backslash (\).

10.2.6 Functions and stored procedures

A script, expression or custom function can call most built-in or custom functions. A function cannot call itself or call another function that would lead to a recursive call. For example, function A cannot call function A, nor can function A call function B if function B calls function A.

You can also use functions and stored procedures imported from a DBMS. Use the exact case for names of imported functions and stored procedures

- Built-in functions are listed in the function wizard and smart editor grouped by category.
- You can find the list of Custom functions in both function wizard and the smart editor.
- Imported functions and stored procedures are listed in the function wizard and smart editor under the datastore used to import them.

Related Information

[Creating a custom function \[page 1243\]](#)

[Importing metadata for stored procedures \[page 1251\]](#)

10.2.7 Operators

The operators you can use in scripts and expressions are listed in the following table, in order of precedence.

Note that when operations are pushed to a DBMS to perform, the precedence is determined by the rules of the DBMS.

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division

Operator	Description
=	Assignment, comparison
<	Comparison, less than
<=	Comparison, less than or equal to
>	Comparison, greater than
>=	Comparison, greater than or equal to
!=	Comparison, not equal to
	Concatenate
%	Return the remainder when one number is divided by another
AND	Logical AND
OR	Logical OR
NOT	Logical NOT
IS NULL	Comparison, is a NULL value
IS NOT NULL	Comparison, is not a NULL value
LIKE	Comparison, matches a specific character string with a specified pattern. You can use the following wildcards with LIKE: <ul style="list-style-type: none"> • %: A string of zero or more characters. • _ (underscore): A single character. • []: A single character with a specific range or set. • [^]: A single character not within the specified range or set.
NOT LIKE	Comparison, excludes rows that match the LIKE criterion.

When you use the smart editor to add operators, you can type in operators or use the built-in key pad. Access the key pad from the smart editor's tool bar.

You can use a comparison in the following ways:

- In a script or script function as a condition; for example:

```
if ($x IN (1,2,3)), while ($x IN (1,2,3)) and ifthenelse()
```

- In a data flow such as in a WHERE clause ifthenelse() function, case transform, etc.
- As a condition of the IF block, WHILE block or TRY CATCH block

The following examples illustrate valid comparison expression syntax:

```
expression = expression
expression != expression
expression < expression
expression > expression
expression <= expression
expression >= expression
expression IS NULL
expression IS NOT NULL
expression IN (expression list)
expression IN domain
expression LIKE constant
```

```
expression NOT LIKE constant
```

NOT (any of the above comparisons); for example NOT (\$x IN (1,2,3))

```
comparison OR comparison  
comparison AND comparison
```

Note that the following syntax is not valid:

```
$x NOT IN (1,2,3)  
EXIST or NOT EXIST
```

10.2.8 NULL values

Indicate NULL values using the keyword NULL.

For example, you can check whether a column (COLX) is null or not:

```
COLX IS NULL  
COLX IS NOT NULL
```

The software does not check for NULL values in data columns. Use the function `nvl` to remove NULL values.

Related Information

[nvl \[page 1177\]](#)

10.2.8.1 NULL values and empty strings

This section discusses the rules for NULL values and empty strings.

The following are the two rules that regulate empty strings.

- When you assign an empty string to a variable of type varchar, the value of the variable will be treated as a zero-length string. An error results if you assign an empty string to a variable that is not a varchar. To assign a NULL value to a variable of any type, use the NULL constant.
- As a constant (' '), the software treats the empty string as a varchar value of zero length. Use the NULL constant for the null value.

i Note

Oracle does not distinguish an empty string from a NULL value. When you insert an empty string or a NULL value into a varchar column, Oracle treats both the empty string and NULL value as NULL values. Therefore, the software treats the value as a NULL value.

The following are the three rules with NULLs and empty strings in conditionals:

- The Equals (=) and Not Equal to (<>) comparison operators against a null value always evaluates to FALSE. This FALSE result includes comparing a variable that has a value of NULL against a NULL constant. The following table shows the comparison results for the variable assignments \$var1 = NULL and \$var2 = NULL:

Condition	Translates to	Returns
If (NULL = NULL)	NULL is equal to NULL	FALSE
If (NULL != NULL)	NULL is not equal to NULL	FALSE
If (NULL = '')	NULL is equal to empty string	FALSE
If (NULL != '')	NULL is not equal to empty string	FALSE
If ('bbb' = NULL)	bbb is equal to NULL	FALSE
If ('bbb' != NULL)	bbb is not equal to NULL	FALSE
If ('bbb' = '')	bbb is equal to empty string	FALSE
If ('bbb' != '')	bbb is not equal to empty string	TRUE
If (\$var1 = NULL)	NULL is equal to NULL	FALSE
If (\$var != NULL)	NULL is not equal to NULL	FALSE
If (\$var1 = '')	NULL is equal to empty string	FALSE
If (\$var != '')	NULL is not equal to empty string	FALSE
If (\$var1 = \$var2)	NULL is equal to NULL	FALSE
If (\$var != \$var2)	NULL is not equal to NULL	FALSE

The following table shows the comparison results for the variable assignments \$var1 = '' and \$var2 = '':

Condition	Translates to	Returns
If (\$var1 = NULL)	Empty string is equal to NULL	FALSE
If (\$var != NULL)	Empty string is not equal to NULL	FALSE
If (\$var1 = '')	Empty string is equal to empty string	TRUE
If (\$var != '')	Empty string is not equal to empty string	FALSE
If (\$var1 = \$var2)	Empty string is equal to Empty string	TRUE
If (\$var != \$var2)	Empty string is not equal to Empty string	FALSE

- Use the IS NULL and IS NOT NULL operators to test the presence of null values. For example, assuming a variable is assigned: \$var1 = NULL;

Condition	Translates to	Returns
If ('bbb' IS NULL	bbb is NULL	FALSE

Condition	Translates to	Returns
If ('bbb' IS NOT NULL)	bbb is not NULL	TRUE
If ('' IS NULL)	Empty string is NULL	FALSE
If ('' IS NOT NULL)	Empty string is not NULL	TRUE
If (\$var1 IS NULL)	NULL is NULL	TRUE
If (\$var IS NOT NULL)	NULL is not NULL	FALSE

- When comparing two variables, always test for NULL. In this scenario, you are not testing a variable with a value of NULL against a NULL constant (as in the first rule). Either test each variable and branch accordingly or test in the conditional as shown in the second row of the following table.

Condition	Recommendation
if(\$var1 = \$var2)	Do not compare without explicitly testing for NULLs. Using this logic is not recommended because any relational comparison to a NULL value returns FALSE.
if (((\$var1 IS NULL) AND (\$var2 IS NULL)) OR (\$var1 = \$var2))	Will execute the TRUE branch if both \$var1 and \$var2 are NULL, or if neither are NULL but are equal to each other.

10.2.9 Debugging and Validation

If you are using the smart editor to create a script or a custom function, select the [Validate](#) icon in the tool bar (or right-click and select [Validate](#)). Errors are listed in the smart editor window under the text box. Double-click each error to see where it occurred in your script.

Otherwise:

- Select the [Debug > Validate > Current View](#) command to check the syntax of expressions used in the definition of the current object. Errors are displayed in the Output window.
- Select the [Debug > Validate > All Objects in View](#) command to check the syntax of expressions used in the current object definition and the definitions of all objects called by the current object. Errors are displayed in the Output window.

By default, Data Services lists ten errors before it aborts parsing. By reporting ten errors for each validation, Data Services allows you to shorten the edit-validate-test cycle. You can repair several errors in each iteration of the cycle.

If you want to change this default setting, select [Tools > Options > Job Server > General](#) and set the options as follows:

Section:	Enter "Parser".
Key:	Enter "NumErrors".
Value:	Enter a positive number.

For each error, Data Services provides a description of the problem.

Data Services also provides three menu options for errors. Right-click an error to use these menu options:

View: The [View](#) option allows you to view the error text in a smaller, separate window for easy reading.

Go To Error: The [Go to Error](#) option places an error icon near the line that caused the error. You can also double-click the error to place this icon. This option not available for all errors, however.

Copy: The [Copy](#) option copies the error text on to the clipboard so you can place it elsewhere.

When you execute a job, you may still encounter errors in an expression that are returned from the operating system or the DBMS. When executing a job, Data Services sends as many operations as possible to the DBMS to evaluate. The DBMS may evaluate part of the expression and return a value that contributes to evaluating the rest of the expression.

10.2.10 Keywords

The following sections describe the keywords in the scripting language.

Keywords are listed in the selection list of the smart editor filtered by the context from which the smart editor is opened.

Related Information

[About Reserved Words \[page 1440\]](#)

10.2.10.1 BEGIN

Indicates the beginning of the code that becomes the function, script, or other construct.

BEGIN and END statements are added automatically to function, transform, and script definitions.

10.2.10.2 CATCH

Indicates a catch for a try/catch block.

If an error occurs while executing any of the statements between the TRY and the CATCH statements, the software executes the statements defined by the CATCH. Use the CATCH keyword as shown in the following script, or use `CATCH (ALL)`.

```
BEGIN
TRY
    BEGIN
        <script_step>;
        <script_step>;
    END
END
```

```
CATCH (<exception_number>)
  BEGIN
    <catch_step>;
    <catch_step>;
  END
CATCH (<exception_number>)
  BEGIN
    <catch_step>;
    <catch_step>;
  END
END
```

10.2.10.3 ELSE

Defines the second branch for an IF statement.

If no ELSE follows the IF, no action is taken if the condition is not met.

10.2.10.4 END

Indicates the end of the code that becomes the function, script, or other construct.

BEGIN and END statements are added automatically to function, transform, and script definitions.

10.2.10.5 IF

Indicates a conditional step in the code.

An IF statement can be constructed with or without an ELSE step. Use the IF keyword as follows:

```
IF (<condition>) <script_step>; ELSE <script_step>;
```

or

```
IF (<condition>) <script_step>;
```

where `<condition>` is an expression that evaluates to True or False. `<script_step>` indicates the set of instructions to execute based on the value of `<condition>`. If `<script_step>` contains more than one statement, enclose these statements in BEGIN and END statements.

Related Information

[RepeatString function \[page 1271\]](#)

10.2.10.6 RETURN

Indicates the value to be returned by a function.

Use the RETURN keyword as follows:

```
RETURN (<expression>);
```

where <expression> defines the value to be returned.

10.2.10.7 TRY

Indicates the start of a try/catch block.

Related Information

[CATCH \[page 1268\]](#)

10.2.10.8 WHILE

Defines a set of statements to execute until a condition evaluates to FALSE.

Use the WHILE keyword as follows:

```
WHILE (<condition>) <script_step>;
```

where <condition> is an expression that evaluates to True or False. <script_step> indicates the set of instructions to execute based on the value of <condition>. If <script_step> contains more than one statement, enclose each statement in BEGIN and END statements.

10.3 Sample scripts

The following examples show how a script would be formatted for the Square and Repeat String functions.

10.3.1 Square function

This function accepts an integer and returns the square of the input value. To define this script, supply the following values to Data Services:

Description	Value	Arguments
Function name	Square	
Return value	Return	int
Input value	input_int	int, input

The text of the function script is as follows:

```
RETURN($input_int * $input_int);
```

10.3.2 RepeatString function

This function accepts a one-character string and an integer indicating the number of times to repeat the input character. It outputs the created string or, if the input character or repeat number is NULL, it outputs a NULL value. It uses a variable to keep track of the string components as it is being generated.

To define this script, supply the following values to Data Services:

Description	Value	Arguments
Function name	RepeatString	
Return value	Return	VARCHAR (255),allow NULL
Input string	InputCharacter	VARCHAR (1), input, allow NULL
Input integer	RepeatNumber	INT, input, allow NULL
Internal variable	Partial	VARCHAR (255), allow NULL

The text of the function is as follows:

```
IF ($InputCharacter = NULL)
BEGIN
    $Partial = NULL;
    RETURN NULL;
END
IF ($RepeatNumber = NULL)
    RETURN NULL;
IF ($RepeatNumber > 255)
    $RepeatNumber = 255;
IF ($RepeatNumber < 0)
    $RepeatNumber = 0;
BEGIN
    WHILE ($RepeatNumber != 0)
    BEGIN
        $Partial = ($Partial || $InputCharacter);
        $RepeatNumber = ($RepeatNumber - 1);
    END
    RETURN $Partial;
END
```

11 Metadata in Repository Tables and Views

Data Services provides full access to the repository metadata used by the applications you create. To access this metadata for application analysis, either manually enter SQL SELECT statements or open the metadata reporting tool.

This section describes the tables and views in the repository that are useful for metadata analysis. This section also provides sample SQL SELECT statements for creating reports using a SQL editor.

For more information, see the *Management Console Guide*.

11.1 Auditing metadata

This set of tables in the Data Services repository stores the statistics that the audit feature collects.

- [AL_AUDIT](#) [page 1272]
- [AL_AUDIT_INFO](#) [page 1273]

11.1.1 AL_AUDIT

This table contains audit information about each data flow execution. The column OBJECT_KEY uniquely identifies a data flow execution.

Column Name	Data Type	Description
OBJECT_KEY	INTEGER	Identifies the audit event.
HISTORY_KEY	INTEGER	Refers to the OBJECT_KEY column in the AL_HISTORY table. Use this referential relationship to obtain history information about operational statistics for the data flow.
DF_LANG_KEY	INTEGER	Refers to the OBJECT_KEY column in the AL_LANG table. Use this referential relationship to obtain the definition for the data flow.
STATUS	INTEGER	Audit status can be one of the following values: <ul style="list-style-type: none">• 0 — Not audited• 1 — Audit rule succeeded• 2 — Audit information collected. This status occurs when you define audit labels to collect statistics but do not define audit rules.• 3 — Audit rule failed
RULEINFO	VARCHAR(255)	The audit rule that failed and the values of the left-hand-side (LHS) and right-hand-side (RHS) of the Boolean expression.

❖ Example

The following query returns the names of data flows that failed an audit and the audit rules that failed.

```
SELECT L.NAME, RULEINFO
FROM AL_AUDIT A, AL_LANG L
WHERE A.STATUS = 3
      AND A.DF_LANG_KEY = L.OBJECT_KEY
```

11.1.2 AL_AUDIT_INFO

This table contains information about the audit statistics.

Column Name	Data Type	Description
AUDIT_KEY	INTEGER	Refers to the OBJECT_KEY column in the AL_AUDIT table. Use this referential relationship to obtain audit information for the data flow.
LABEL	VARCHAR(255)	Refers to the OBJECT_KEY column in the AL_HISTORY table. Use this referential relationship to obtain history information for the data flow.
VALUE	VARCHAR(255)	Value of the label. This value can be one of the following: <ul style="list-style-type: none">• Number of rows in a table or whole row set• Sum of the values in a column• Average of the values in a column• Checksum of the values in a column

❖ Example

The following query returns the values of the labels of audit points that failed.

```
SELECT L.NAME DF_NAME, B.LABEL, B.VALUE AUDIT_VALUE
FROM AL_AUDIT A, AL_AUDIT_INFO B, AL_LANG L
WHERE A.OBJECT_KEY = B.AUDIT_KEY
      AND A.STATUS = 3
      AND A.DF_LANG_KEY = L.OBJECT_KEY;
```

If you want to see values of audit labels for a specific data flow, use a query similar to the following:

```
SELECT L.NAME DF_NAME, B.LABEL, B.VALUE AUDIT_VALUE
FROM AL_AUDIT A, AL_AUDIT_INFO B, AL_LANG L
WHERE A.OBJECT_KEY = B.AUDIT_KEY
      AND A.STATUS = 3
      AND A.DF_LANG_KEY = L.OBJECT_KEY
      AND L.NAME = 'Validation_DF';
```

11.2 Imported metadata

You can use tables (prefixed by AL_) and views (prefixed by ALVW_) to capture information about the metadata imported into SAP Data Services from external databases and applications (such as Oracle, PeopleSoft, and SAP Applications).

11.2.1 AL_INDEX

This table contains index information for tables.

Column Name	Description
TABLEKEY	The table ID associated with the index. The TABLEKEY relates to table information in ALVW_TABLEINFO.
NAME	The index name.
COLNAME	The name of the table's index column.
COLPOSITION	Position of the column in the index.

❖ Example

The following query returns the index information associated with a table EMPLOYEE in datastore ORA_DS:

```
SELECT NAME, COLNAME, COLPOSITION
FROM AL_INDEX, ALVW_TABLEINFO
WHERE AL_INDEX.TABLEKEY =
      ALVW_TABLEINFO.TABLE_ID
      AND TABLE_NAME = 'EMPLOYEE'
      AND DATASTORE = 'ORA_DS'
```

11.2.2 AL_PCOLUMN

This table contains partition information for tables.

Column Name	Description
TABLEKEY	The partitioned table ID. The TABLEKEY relates to table information in ALVW_TABLEINFO.
COLNAME	The name of the table's partition column.

Column Name	Description
COLPOSITION	Position of the column in the partition.

11.2.3 AL_PKEY

This table contains primary key information for tables.

Column Name	Description
TABLEKEY	The table ID associated with a primary key. The TABLEKEY relates to table information in ALVW_TABLEINFO.
COLNAME	Name of the table's primary key column.
COLPOSITION	Position of the primary key column.

❖ Example

The following query returns the primary key associated with a table EMPLOYEE in datastore ORA_DS:

```
SELECT COLNAME, COLPOSITION
FROM AL_PKEY, ALVW_TABLEINFO
WHERE AL_PKEY.TABLEKEY =
      ALVW_TABLEINFO.TABLE_ID
      AND TABLE_NAME = 'EMPLOYEE'
      AND DATASTORE = 'ORA_DS'
```

11.2.4 ALVW_COLUMNATTR

This view contains information about column attributes.

Column Name	Data type	Description
TABLE_NAME	Varchar(256)	Name of the table.
TABLE_OWNER	Varchar(256)	Owner of the table.
DATASTORE	Varchar(256)	The datastore to which this table belongs.
COLUMN_NAME	Varchar(256)	Name of the column.

Column Name	Data type	Description
COLUMN_ATTR	Varchar(64)	Name of the attribute (property of this table). The name of the attribute will be the same as seen when viewing the attributes of a table in the Designer's interface. Examples: Description, Business_Name, Date_last_loaded, Date_created, Total_Number_Of_Rows_Processed.
COLUMN_ATTR_VALUE	Varchar(255)	Value of this attribute.

❖ Example

The following query returns all columns and their descriptions for table EMP in datastore HR:

```
SELECT COLUMN_NAME, COLUMN_ATTR_VALUE
FROM ALVW_COLUMNATTR
WHERE TABLE_NAME = 'EMP'
      AND COLUMN_ATTR = 'Description'
      AND DATASTORE = 'HR'
```

11.2.5 ALVW_COLUMNINFO

This view contains information about the columns in a table.

Column Name	Data type	Description
TABLE_NAME	Varchar(256)	Name of the table.
TABLE_OWNER	Varchar(256)	Owner of the table.
DATASTORE	Varchar(256)	The datastore to which this table belongs.
COLUMN_NAME	Varchar(256)	Name of the column.
COLUMN_DATATYPE	Varchar(20)	Data type of this column. Examples: integer, datetime, decimal, real.
COLUMN_LENGTH	Int	Length of this column in bytes.

Column Name	Data type	Description
COLUMN_PRECISION	Int	Precision of this column (applies only to columns with the data type decimal).
COLUMN_SCALE	Int	Scale for this column (applies only to columns with the data type decimal).
COLUMN_IS_NULLABLE	Int	One (1) indicates this column will accept NULL values, 0 indicates not.
COLUMN_ID	Int	The ID for this column within the repository. It can be used to join with other tables containing additional column information.

❖ Example

The following query returns all columns in table EMP in datastore HR:

```
SELECT COLUMN_NAME
FROM ALVW_COLUMNINFO
WHERE TABLE_NAME = 'EMP'
AND DATASTORE = 'HR'
```

11.2.6 ALVW_FKREL

This view contains information about the primary and foreign key relationships among tables.

Column Name	Description
TABLEKEY	The table ID.
PKCOLUMN	Primary key column in this table.
FKTABLE	The table where this column is referenced.
FKCOLUMN	The column in the 'foreign' table which is the same as the primary column.

❖ Example

The following query returns the primary and foreign key information associated with table <EMPLOYEE> in datastore <ORA_DS>.

```
SELECT FKTABLE, FKCOLUMN, PKCOLUMN
FROM ALVW_FKREL, ALVW_TABLEINFO
WHERE ALVW_FKREL.TABLEKEY =
      ALVW_TABLEINFO.TABLE_ID
```

```
AND TABLE_NAME='EMPLOYEE'  
AND DATASTORE='ORA_DS'
```

11.2.7 ALVW_MAPPING

The ALVW_MAPPING view joins the AL_COLMAP and the AL_COLMAP_TEXT tables. These tables contain information about target tables and columns, the sources used to populate target columns, and the transforms Data Services applies to sources before applying them to targets. Data Services uses the ALVW_MAPPING view for impact analysis in Metadata Reports.

The column mapping calculation generates the following information for target columns:

- The source column(s) from which the target column is mapped.
- The expressions used to populate target columns.

Data Services stores column mappings of nested source and target data in data flows using both the ALVW_MAPPING view and the AL_COLMAP_NAMES table.

i Note

For those column names with varchar(256), the maximum length depends on the Data Services repository type. For most repository types the maximum length is 256, for MySQL the length is 64, and for MS SQL server the length is 128.

ALVW_MAPPING view

Column Name	Data type	Description
DF_NAME	varchar(256)	Data flow that populates the target table.
TRG_TAB_NAME	varchar(256)	Name of the target table.
TRG_TAB_ID	int	ID for this table within the repository.
TRG_TAB_DESC	varchar(100)	Description of the target table.
TRG_OWNER	varchar(256)	Owner of the target table.
TRG_DS	varchar(256)	Datastore of the target table.
TRG_TYPE	varchar(64)	Type of target. Examples: table, BW transfer structure.
TRG_USAGE	varchar(65)	Usage of the target table. Examples: fact, dimension, lookup. Currently set to NULL.
TRG_COL_NAME	varchar(256)	Column name in the target.
TRG_COL_ID	int	ID for this column in the repository.
TRG_COL_DESC	varchar(100)	Description of this column.
SRC_TAB_NAME	varchar(256)	Name of the source table used to populate the target.
SRC_TAB_ID	int	ID for this table within the repository.
SRC_TAB_DESC	varchar(100)	Description of the source table.
SRC_OWNER	varchar(256)	Owner of the source table.

Column Name	Data type	Description
SRC_DS	varchar(256)	Datastore of the source table.
SRC_TYPE	varchar(64)	Type of source. Examples: table, file.
SRC_COL_NAME	varchar(256)	Name of the source column.
SRC_COL_ID	int	ID for this column in the repository.
SRC_COL_DESC	varchar(100)	Description of this column.
MAPPING_TYPE	varchar(65)	Types of source to target mapping. Examples: direct, computed, lookup.
MAPPING_TEXT	varchar(255)	The expression used to map the source to the target column.

Related Information

[Storing nested column-mapping data \[page 1281\]](#)

11.2.7.1 Example use case

The following query returns target tables and columns populated from the column `EMPID` in table `EMP` (in datastore `HR`):

```
SELECT TRG_TAB_NAME, TRG_COL_NAME
FROM ALVW_MAPPING
WHERE SRC_TAB_NAME = 'EMP'
      AND SRC_COL_NAME = 'EMPID'
      AND SRC_DS = 'HR'
```

11.2.7.2 Mapping types

The `AL_COLMAP_TEXT` table contains information qualifying the mapping relationships. This information, stored in the `MAPPING_TYPE` column, can have the following values:

Mapping Type	Description
Direct	<p>The target column is mapped directly from a source column with no expression to transform it.</p> <p>For example, <code>EMPID</code> (employee ID) mapped directly from source to target.</p>

Mapping Type	Description
Computed	<p>There is an expression associated with the target column.</p> <p>For example, NAME is <code>LAST_NAME ', ' FIRST_NAME</code>.</p>
Generated	<p>There is no source column associated with the target column.</p> <p>For example, the target table is mapped to a constant or a function, such as <code>sysdate</code>, or is obtained from a transform, such as <code>Date_Generation</code>.</p>
LookedUp	A lookup function is used in the expression.
Merged	<p>Two data streams are merged to populate the target table.</p> <p>The two expressions mapped to the target table are separated by AND.</p>
Not mapped	The column in the target table is not being populated by the data flow.
Unknown	Data Services is unable to identify the expression used to map the target column. This happens only under unusual error conditions.

11.2.7.3 How mappings are computed

When a data flow processes information, it performs potentially complex transformations on data in preparation for loading it into one or more target tables. Typical operations include:

- Reading data from the appropriate sources.
- Processing data using Query transforms or other transforms.
- Splitting the data stream and then merging it again.

Consider the following example, where two transformations operate against a value from one column of a source table.

The information is captured in `AL_COLMAP_TEXT` as follows:

Target column	Source column	Mapping expression
Total_value	Price	<code>((Price x 1.17) x 112)</code>

This kind of information becomes more valuable as transformation complexity increases. Consider the following example:

- Data flow `DF_1` reads three columns (a, b, c) from source table S.

- Table S is connected to a Query transform Q1.
- The Query transform output has four columns (Qa, Qb, Qc, and Qd) whose mapping expressions are S.a, S.b, S.c and S.a – S.b.
- The output of Q1 is connected to Query transform Q2, which has two columns Q2y and Q2z whose expressions are Qa – Qb and Qc – Qd.
- The output of Q2 is loaded into target table T, which has two columns: T1 and T2.

The mapping expressions for target columns T1 and T2 are computed by starting from the end point (the target object) and "walked" back through the list of transforms, with columns of a transform written in terms of expressions from the previous transform.

When processing is started on data flow DF_1, it starts with column T1 of target table T.

The expression for T1 is Q2y, which in turn is A – dB, which can be written as S.a – S.b. Therefore the mapping expression is S.a – S.b and column T1 has two source columns—it is mapped from S.a and S.b. The AL_COLMAP table contains two rows for the target column to describe the two source columns.

In the case of T2, it is mapped from DC – JD, which can be written as S.c – (S.a – S.b). In this case, there are three rows for this target column in the AL_COLMAP table, one for each source column.

11.2.7.4 Mapping complexities

If a data flow calls another data flow and then loads a target table, the mappings are expressed in terms of the tables and columns used within the other data flow. Information is generated by "drilling down" into the other data flow to continue the mapping process.

The situation in which the Merge transform is used within a data flow is a bit more complex, because when two data streams are merged, there are two ways to populate a target table. This possibility is captured by separating the mapping expressions with the keyword **AND**. For example, a target column could be populated from `S.a AND R.a`.

Transforms like Hierarchy_Flattening and Pivot also introduce column mapping complexities.

It is also possible that some target columns are mapped by constants or expressions that do not use source columns. In this case there will be no rows in the AL_COLMAP table for the target column. The mapping expression in the AL_COLMAP_TEXT table will reflect this.

If a target column is populated with a call to the lookup function, then its source columns are both the looked up column and the key used to do the lookup.

11.2.7.5 Storing nested column-mapping data

SAP Data Services calculates column mappings (identifies the source column(s) and expressions in use to map the target column) for all data flows including those that use nested data.

The following objects and conditions are supported:

- XML files or messages
- IDOC files or messages

- Custom and adapter functions
- SAP Applications and PeopleSoft hierarchies
- Column mappings that perform nesting or un-nesting (target columns mapped from a nested or un-nested data set)
- Nested columns used as parameters in custom or adapter functions (including SAP Application RFC output parameters, BAPI function calls, and database stored procedures)
- Embedded data flows
- ABAP data flows
- Correlated columns

You can map a column in a nested schema from a column in the input schema of the nested schema, or from a column in the input schema of the parent (or any ancestor) of the nested schema. If you map a column from an ancestor, the column is correlated.

Transforms support nested column-mapping data as follows:

- Query transforms process nested data and mappings are stored in repository tables and views.
- The software allows nested column mappings to pass through the Merge, Case, and Map_Operation transforms.
- Other transforms do not process nested data.

Nested (NRDM) notations that represent column names are longer than those used for a flat schema column.

- A column in a flat schema is represented by Table. Column, for example, "mats_emp.empno". Note that Table may represent a database table, a file, an XML message or file, an IDOC message or file, and so on.
- A column in a nested schema is represented by

```
Table.Subschema1...SubschemaN.Column
```

For example, "personnel.name.given" represents a column of a nested schema which has three components. The first component is the Table. The last component is the Column. The middle components identify the nested levels in the Table.

Because the TRG_COL_NAME and SRC_COL_NAME columns in the ALVW_MAPPING view of the repository are VARCHAR(64) (MySQL only) and not big enough to store long NRDM column names, the software uses the AL_COLMAP_NAMES table to support nested data.

AL_COLMAP_NAMES table

Column Name	Data type	Description
DF_NAME	varchar(256)	Data flow with that populates a target table.
COL_ID	varchar(65)	The software generates this value using the following format when a nested column is encountered. <div style="background-color: #f0f0f0; padding: 5px; margin: 10px 0;"> "__DI_NESTED_COLNAME_n" </div> where n is an integer that starts from 1
COL_NAME	varchar(256)	If the software generates a COL_ID value, this column stores the original nested column name.

Column Name	Data type	Description
SEQNUM	int	The software generates this value if more than one set of 256 characters is required to store data in the COL_NAME column. For each set of 256 characters, it generates a new row and a sequence number.

The AL_COLMAP_NAMES table uses the DF_NAME, COL_ID, SEQNUM columns as primary keys. The DF_NAME and COL_ID columns are keyed to the following columns in the ALVW_MAPPINGS view.

- DF_Name is keyed to DF_Name.
- COL_ID is keyed to SRC_COL_NAME and TRG_COL_NAME

The AL_COLMAP_NAMES table also provides an internal mapping mechanism from COL_ID column to COL_NAME.

For example, if a source column BOOKS.AUTHOR.FIRST_NAME is mapped into a target column BOOK.AUTHOR_NAME (an un-nesting is probably in place), you can create a report to query the following column values in the repository:

ALVW_MAPPING view	Column value
TRG_TAB_NAME	BOOK
TRG_COL_NAME	AUTHOR_NAME
SRC_TAB_NAME	BOOKS
SRC_COL_NAME	__DI_NESTED_COLNAME_1
MAPPING_TEXT	BOOKS.AUTHOR.FIRST_NAME

AL_COLMAP_NAMES table	Column value
COL_ID	__DI_NESTED_COLNAME_1
COL_NAME	AUTHOR.FIRST_NAME

The TRG_COL_NAME or SRC_COL_NAME columns in the ALVW_MAPPING view store the COL_ID, if the target or source column is nested. To get the actual column name, lookup the AL_COLMAP_NAMES table using the DF_Name, COL_ID, and COL_NAME.

Flat or un-nested target or source column names are stored using the format Column in TRG_COL_NAME and SRC_COL_NAME. For example, of the three source columns shown below, only the second one is nested:

SRC_COL_NAME
EMPNO
_DI_Nested_Names_1
ENAME

The second value is the only one for which the software generates a column ID. To find this source column's real name, create a report that looks up its COL_NAME from the AL_COLMAP_NAMES table.

11.2.8 ALVW_TABLEATTR

This view contains information about the attributes in a table.

Column Name	Data type	Description
TABLE_NAME	Varchar(256)	Name of the table.
TABLE_OWNER	Varchar(256)	Owner of the table. For SAP applications, the value is NULL.
DATASTORE	Varchar(256)	The datastore to which this table belongs.
TABLE_ATTR	Varchar(64)	Name of the attribute (property of this table). The name of the attribute will be the same as seen when viewing the attributes of a table in the Designer's interface. Examples: Description, Business_Name, Date_last_loaded, Date_created, Total_Number_Of_Rows_Processed.
TABLE_ATTR_VALUE	Varchar(255)	Value of this attribute.

❖ Example

The following query returns when table EMP in datastore HR was last loaded:

```
SELECT TABLE_ATTR_VALUE
FROM ALVW_TABLE_ATTR
WHERE TABLE_NAME = 'EMP'
      AND TABLE_ATTR = 'Date_last_loaded'
      AND DATASTORE = 'HR'
```

11.2.9 ALVW_TABLEINFO

This view contains the list of tables imported into the repository.

Column Name	Data type	Description
TABLE_NAME	Varchar(256)	Name of the table.
TABLE_OWNER	Varchar(256)	Owner of the table. For SAP applications, this table is NULL.
DATASTORE	Varchar(256)	The datastore to which this table belongs.

Column Name	Data type	Description
TABLE_ID	int	Internal ID of the table within the repository.

11.3 Internal metadata

You can use this set of tables and views to capture information about built-in metadata objects used by Data Services and the relationships between those objects.

11.3.1 AL_LANG

This table contains various Data Services objects. These objects are also displayed in Data Services' object library.

Column Name	Description
OBJECT_KEY	Internal ID of the object.
OBJECT_TYPE	Type of object (an integer). Query the AL_REPO- TYPE_NAMES table to find the corresponding name for this type. Examples: data flow, table, datastore.
NAME	Name of the object.
VERSION	Indicates the number of times this object has been updated.
TYPE	Subtype of the object.
OWNER	For table objects, the owner.
DATASTORE	For table objects, the datastore to which they belong.
NORMNAME	Unique name for this object in this table.

Example

The following query returns all the data flows in the repository:

```
SELECT OBJECT_KEY, NAME
  FROM AL_LANG A
 WHERE OBJECT_TYPE = 1
    AND VERSION =
      (SELECT MAX(VERSION) FROM AL_LANG B
       WHERE A.NORMNAME = B.NORMNAME)
```

11.3.2 AL_LANGXMLTEXT

This table contains various Data Services objects. These objects are also displayed in Data Services' object library.

Column Name	Description
OBJECT_KEY	Internal ID of the object.
DATE_MODIFIED	Date the object was last modified.
OBJECT_TYPE	Type of object (an integer). Query the AL_REPO- TYPE_NAMES table to find the corresponding name for this type. Examples: data flow, table, datastore.
OBJECT_SUBTYPE	Subtype of the object.
OBJECT_NORMNAME	Unique name for this object in this table.
SEQNUM	Used reorder XML segments.
TEXT_VALUE	Contains the XML representation of the object.

11.3.3 AL_ATTR

This table contains attributes of repository objects provided by Data Services. For example, a description is an object attribute.

Column Name	Description
PARENT_OBJID	The internal ID of an object.
PARENT_OBJ_TYPE	Type of the object (integer).
ATTR_NAME	The attribute name for this object.
ATTR_VALUE	Value of the attribute.

❖ Example

The following query returns all the data flows and their descriptions in the repository:

```
SELECT OBJECT_KEY, NAME, ATTR_VALUE
```



```

FROM AL_LANG O, AL_ATTR A
WHERE O.OBJECT_TYPE = 1
      AND O.OBJECT_KEY = A.PARENT_OBJID
      AND ATTR_NAME = 'Description'
      AND O.VERSION =
        (SELECT MAX(VERSION) FROM AL_LANG B
         WHERE O.NORMNAME = B.NORMNAME)

```

11.3.4 AL_SETOPTIONS

This table includes option settings for all objects.

Column Name	Data type	Description
PARENT_OBJ_ID	numeric(38)	ID of the parent object to which this option belongs.
PARENT_OBJ_TYPE	numeric(38)	Type of the parent object. Examples: batch job or data flow.
CALL_OBJID	varchar(100)	ID of the calling object. (Future use)
CALL_OBJTYPE	varchar(255)	Type of the calling object. (Future use)
OPTION_NAME	varchar(100)	Name of the option.
OPTION_VALUE	varchar(255)	Value of the option.
OVERFLOW_KEY	numeric(38)	KEY value pointing to a row in the AL_OVERFLOW_ATTR table. When an OPTION_VALUE exceeds 255 characters, Data Services adds the remaining characters to AL_OVERFLOW_ATTR and stores the row ID as OVERFLOW_KEY in the AL_SETOPTIONS table.

11.3.5 AL_USAGE

This table is identical to ALVW_PARENT_CHILD except it captures the entire call hierarchy. For example, if a table is used in a data flow which is called from a work flow, then rows appear in this table that associate the work flow (parent) to the table (descendent). The Depth column is unique to this table.

i Note

You need to populate this table explicitly by using the [Calculate Usage Dependencies](#) command.

To query this table, use the following [Example use cases \[page 1288\]](#).

Column Name	Data type	Description
PARENT_OBJ	varchar(64)	Name of the calling object.
PARENT_OBJ_TYPE	varchar(64)	Type of the object. Examples: batch job or data flow.
PARENT_OBJ_DESC	varchar(255)	The description associated with this object.
PARENT_OBJ_KEY	int(4)	Key in the AL_LANG table of the parent object.
DESCEN_OBJ	varchar(64)	Name of the descendant object. For transforms, the name of the output schema of the transform call (if the name of the transform is unique). If it is not unique, Data Services generates a unique numeric suffix and appends that to the given name.
DESCEN_OBJ_TYPE	varchar(64)	Type of the called object. Examples: data flow, table, function, file.
DESCEN_OBJ_DESC	varchar(255)	Description associated with the called object.
DESCEN_OBJ_USAGE	varchar(20)	Applies only to tables and files. How the child is used. Examples: source, target, lookup table.
DESCEN_OBJ_KEY	int(4)	Key in AL_LANG of the descendant object.
DESCEN_OBJ_DS	varchar(64)	Applies only to tables and files. The datastore of the child object.
DESCEN_OBJ_OWNER	varchar(256)	Owner of the child table.
DEPTH	int	Indicates the number of levels between objects in a parent/descendent relationship.

11.3.5.1 Example use cases

You can query the AL_USAGE table using SQL statements. For example, the following report shows a few columns and rows from AL_USAGE listing the objects that are related to the SALES_ORDER target table:

Parent_Obj	Parent_Obj_Type	Descen_Obj	Descen_Obj_Type	Descen_Obj_Desc	Descen_Obj_Usage
Build_Fact	data flow	SALES_ORDER	Table	Sales order target fact table	Target
Daily_Job	work flow	SALES_ORDER	Table	Sales order target fact table	Target
DF_NewOrders	Job	SALES_ORDER	Table	Sales order target fact table	Target

Parent_Obj	Parent_Obj_Type	Descen_Obj	Descen_Obj_Type	Descen_Obj_Desc	Descen_Obj_Usage
Get_IDoc	data flow	SALES_ORDER	Table	Sales order target fact table	Target
IDoc_job	data flow	SALES_ORDER	Table	Sales order target fact table	Target
initial_Job	Job	SALES_ORDER	Table	Sales order target fact table	Target

❖ Example

To find out which jobs load a table

The following query returns which jobs load a particular target:

```
SELECT PARENT_OBJ
FROM AL_USAGE
WHERE PARENT_OBJ_TYPE = 'Job'
      AND DESCEN_OBJ_TYPE = 'Table'
      AND DESCEN_OBJ_USAGE = 'Target'
      AND DESCEN_OBJ = '<targetTable>'
```

❖ Example

To find out which objects depend on a source

The following query returns which objects would be affected if you drop a source table:

```
SELECT PARENT_OBJ, PARENT_OBJ_TYPE
FROM AL_USAGE
WHERE DESCEN_OBJ_TYPE = 'Table'
      AND DESCEN_OBJ = '<targetTable>'
```

❖ Example

To produce a "where used" report for an object

The following query returns what objects call a given object. The given object in this case is the target table SALES_ORDER.

```
SELECT
    AL_USAGE.PARENT_OBJ,
    AL_USAGE.PARENT_OBJ_TYPE,
    AL_USAGE.DECEN_OBJ,
    AL_USAGE.DECEN_OBJ_DESC,
    AL_USAGE.DECEN_OBJ_TYPE,
    AL_USAGE.DECEN_OBJ_USAGE
FROM AL_USAGE
WHERE (AL_USAGE.DECEN_OBJ_TYPE = 'Table'
      AND AL_USAGE.DECEN_OBJ='SALES_ORDER')
```

The following table shows the result for an example repository:

Parent_Obj	Parent_Obj_Type	Descen_Obj	Descen_Obj_Type	Descen_Obj_Desc	Descen_Obj_Usage
Build_Fact	data flow	SALES_ORDER	Table	Sales order target fact table	Target
Daily_Job	work flow	SALES_ORDER	Table	Sales order target fact table	Target
DF_NewOrders	Job	SALES_ORDER	Table	Sales order target fact table	Target
Get_IDoc	data flow	SALES_ORDER	Table	Sales order target fact table	Target
IDoc_job	data flow	SALES_ORDER	Table	Sales order target fact table	Target
initial_Job	Job	SALES_ORDER	Table	Sales order target fact table	Target

❖ Example

To show which jobs load what targets

The following query returns all of the jobs in the repository and what targets they load:

```
SELECT
    AL_USAGE.PARENT_OBJ,
    AL_USAGE.PARENT_OBJ_TYPE,
    AL_USAGE.DSCEN_OBJ,
    AL_USAGE.DSCEN_OBJ_DESC,
    AL_USAGE.DSCEN_OBJ_TYPE,
    AL_USAGE.DSCEN_OBJ_USAGE
FROM AL_USAGE
WHERE (AL_USAGE.PARENT_OBJ_TYPE = 'Job'
      AND AL_USAGE.DSCEN_OBJ_TYPE = 'Table'
      AND AL_USAGE.DSCEN_OBJ_USAGE = 'Target')
```

This table shows the results for an example repository:

Parent_Obj	Parent_Obj_Type	Descen_Obj	Descen_Obj_Type	Descen_Obj_Desc	Descen_Obj_Usage
Daily_Job	Job	SALES_ORDER	Table	Sales order target fact table	Target
IDoc_job	Job	SALES_ORDER	Table	Sales order target fact table	Target
initial_Job	Job	SALES_ORDER	Table	Sales order target fact table	Target

11.3.6 ALVW_FUNCINFO

This view contains a list of functions you defined in Data Services and those you imported into its repository.

Column Name	Description
FUNC_KEY	Internal ID for the function within the repository.
FUNC_NAME	The function name.
FUNC_OWNER	Applies only to imported functions. Owner of the function.
DATASTORE	Applies only to imported functions. Datastore to which the function belongs.

11.3.7 ALVW_PARENT_CHILD

This view contains information about objects (parents) that contain (or call) other objects (children).

Column Name	Description
PARENT_OBJ	Name of the calling object.
PARENT_OBJ_TYPE	Type of the object. Examples: batch job, real-time job, data flow.
PARENT_OBJ_DESC	The description associated with this object.
DESCEN_OBJ	Name of the called object.
DESCEN_OBJ_TYPE	Type of the called object. Examples: data flow, table, function, file.
DESCEN_OBJ_DESC	Description associated with the called object.
DESCEN_OBJ_USAGE	Applies only to tables and files. How the child is used. Examples: source, target, lookup table.
DESCEN_OBJ_DS	Applies only to tables and files. The datastore of the child object.
DESCEN_OBJ_OWNER	Owner of the child table.

Example

You can query this table to:

- View which data flows load table EMP

```
SELECT PARENT_OBJ
FROM ALVW_PARENT_CHILD
WHERE DESCEN_OBJ_TYPE = 'Table'
      AND DESCEN_OBJ = 'EMP'
      AND DESCEN_OBJ_USAGE = 'Target'
```

- View the data flows called by job HR_INITIAL_LOAD

```
SELECT DESCEN_OBJ
FROM ALVW_PARENT_CHILD
WHERE PARENT_OBJ_TYPE = 'Job'
      AND PARENT_OBJ = 'HR_INITIAL_LOAD'
      AND DESCEN_OBJ_TYPE = 'Dataflow'
```

11.4 Operational metadata

The AL_HISTORY table and the ALVW_FLOW_STAT view contain information about the run-time statistics of Data Services jobs and data flows.

11.4.1 AL_HISTORY

This table contains information about the execution statistics of jobs.

Column Name	Description
OBJECT_KEY	Internal ID of the job within the repository.
INST_MACHINE	Computer on which the job was executed.
TYPE	Batch job or a real-time job.
SERVICE	Name of the job.
START_TIME	Time when the job was launched.
END_TIME	Time when the job completed.
EXECUTION_TIME	Difference between start time and end time in seconds.
STATUS	Status of the job upon completion. Examples: Error (E), Success (S).
HAS_ERROR	Displays a zero if there are no errors.

❖ Example

The following query returns the run statistics of all successfully executed jobs:

```
SELECT SERVICE, INST_MACHINE, START_TIME, END_TIME, EXECUTION_TIME
FROM AL_HISTORY A
WHERE HAS_ERROR = 0
      AND SERVICE NOT IN ('di_job_al_mach_info', 'CD_JOB_d0cafae2')
      AND OBJECT_KEY = (SELECT MAX(OBJECT_KEY) FROM AL_HISTORY b WHERE
A.SERVICE = B.SERVICE)
```

11.4.2 DQVW_AGR_RPTSTATS

This view contains aggregated data quality statistics data.

Column	Description
SERVICE_ID	
JOB_KEY	Identifier that represents a job run. Whenever a job is executed, a new ID is given.
OBJECT_ID	GUID (globally unique identifier) assigned to a transform.
OBJECT_NAME	Name of the object in the data flow.
JOB_NAME	Name of the executed job.
OBJECT_TYPE	Type of the object in the data flow.
DATAFLOW_NAME	Name of the data flow.
PATHS	Position of the object in the data flow.
ROW_COUNT	Number of rows in object.
EXECUTION_TIME	Difference between start time and end time in seconds.
START_TIME	Time when the job was launched.
END_TIME	Time when the job completed.

11.4.3 ALVW_FLOW_STAT

This view contains information about the execution statistics of transforms within data flows.

Column Name	Description
DATAFLOW_NAME	Name of the data flow.
JOB_NAME	Name of the executed job.
JOB_KEY	Identifier that represents a job run. Whenever a job is executed, a new ID is given.
JOB_RUNID	Identifier that represents a single job over its duration. For example, if a job is set to recover from a failed execution and then the job fails, it would restart with the same JOB_RUNID.
RUN_SEQ	Unique identifier for a particular sequence of an execution.
PATH	Position of the transform in the data flow.
OBJECT_NAME	Name of the transform.
OBJECT_TYPE	Type of the transform.
ROW_COUNT	Number of rows processed by this transform.
START_TIME	Time when transform started executing.
END_TIME	Time when transform stopped.
EXECUTION_TIME	The difference between start and end time.

11.4.4 DQVW_REPORTS_STAT

This view contains information about report statistics for jobs.

Column	Description
JOB_NAME	Name of the executed job.
JOB_KEY	Identifier that represents a job run. Whenever a job is executed, a new ID is given.
OBJECT_ID	GUID (globally unique identifier) assigned to a transform.
JOB_RUNID	Identifier that represents a single job over its duration. For example, if a job is set to recover from a failed execution and then the job fails, it would restart with the same JOB_RUNID.
PATH	Position of the object in the data flow.
OBJECT_NAME	Name of the object in the data flow.
OBJECT_TYPE	Type of the object in the data flow.
ROW_COUNT	Number of rows in object.
START_TIME	Time when the job was launched.
END_TIME	Time when the job completed.

Column	Description
EXECUTION_TIME	Difference between start time and end time in seconds.
DATAFLOW_NAME	Name of the dataflow.

12 Data quality reports and statistics tables

There are two types of data quality repository tables: regulatory and informative tables and reports, and data quality statistics tables.

Data quality tables in the repository contain job processing statistics.

Some transforms create automatic reports when you enable reports in the transform editor. The software creates the reports based on the availability of specific statistics in repository tables. These reports may be regulatory, like the AMAS form for Australia address matching processing, or informative, like the U.S. Addressing Report. The following list contains the regulatory and informative tables:

- Data Cleanse
- DSF2 Walk Sequencer
- Geocoder
- Global Address Cleanse
- Match
- USA Regulatory Address Cleanse

The data quality statistic tables populated by the Data Cleanse, Global Address Cleanse, and Geocoder transforms identify significant cleanse process changes based on the entities. The data quality statistics tables contain summary and record-level statistics.

The following list contains the data quality statistics tables that include information about the assignment process:

- CLEANSE_ADDRESS_RECORD_INFO_ (available for Global Address Cleanse)
- CLEANSE_CHANGE_INFO_ (available for Data Cleanse and Global Address Cleanse)
- CLEANSE_COMPONENT_INFO_ and the TASK_COLUMN_DEFINITIONS_ (available after 4.2.6.0 release for Data Cleanse and Global Address Cleanse)
- CLEANSE_INFO_CODES_ (available for Data Cleanse and Global Address Cleanse)
- CLEANSE_STATISTICS_ (available for Data Cleanse and Global Address Cleanse)
- GEOCODE_INFO_CODES_ (available for Geocoder)
- GEOCODE_STATISTICS_ (available for Geocoder)

The TASK_LOCALIZATION files available for download provide the descriptions for the data included in some of the columns.

Enable statistics in the applicable transform option editor. The software populates the corresponding repository tables with statistics when you run your job.

For information about controlling the history and log file retention periods, see the topic “History file and log file retention” in the *Administrator Guide*.

Related Information

[Data quality statistics tables and supplemental content information \[page 1344\]](#)

[Repository tables and related reports \[page 1297\]](#)

[Match repository statistics tables \[page 1325\]](#)

12.1 Repository tables common columns

The following table contains descriptions for columns that are common to most of the repository tables and the data quality statistics tables.

These columns, combined with a unique column from the table, make up the primary key for each table. These columns contain platform-generated data. Each repository table contains these columns unless indicated in the individual description.

Column	Data type definition	Description
OBJECT_KEY	INT	Identification for a specific run (also called Run ID). May not appear in each table.
OBJECT_ID	NVARCHAR (255)	GUID (globally unique identifier) assigned to a transform. Appears in each table.
ROW_ID	INT	An internal identification generated by the transform that uniquely identifies a row processed by that transform. Appears in each non-summary data quality statistics table.
TABLE_ID	INT	An internal identification generated by the transform that uniquely identifies the source table.

12.2 Repository tables and related reports

The software generates many reports automatically when you enable them in the transforms.

Enable various reports in the USA Regulatory Address Cleanse, DSF2 Walk Sequencer, Global Address Cleanse, and Match transforms. The following table lists the transform-generated report names, the corresponding repository table name, and the applicable transforms.

Report	Repository table	Transform(s)
US CASS report: USPS Form 3553	PSFORM3553DATA	USA Regulatory Address Cleanse

Report	Repository table	Transform(s)
NCOALink Processing Summary report	NCOALINKSUMMARY MEDSTATS NCOACERTIFICATIONDATA	USA Regulatory Address Cleanse
Customer Service Log	CSLSTATS	USA Regulatory Address Cleanse
Broker and List Administrator file	PAFBALASTATS	USA Regulatory Address Cleanse
Delivery Sequence Invoice report	DSFSEQUENCESTATS	DSF2 Walk Sequencer
US Addressing Report	DPVLACSLINKSUMMARY	USA Regulatory Address Cleanse
US Regulatory Locking Report	USREGULATORYLOCKING	USA Regulatory Address Cleanse
Canada SERP report: Statement of Address Accuracy	SERPADDRACCURACY	Global Address Cleanse
Australia AMAS report: Address Matching Processing Summary	AMASADRPROCSUMMARY	Global Address Cleanse
Address Information Code Sample report	ADDRINFOCODEDATA	Global Address Cleanse USA Regulatory Address Cleanse
Address Information Code Summary report	ADDRINFOCODESUMMARY ADDRSTATUSCODEDATA	Global Address Cleanse USA Regulatory Address Cleanse
Address Validation Summary report	ADDRVALIDATESUMMARY	Global Address Cleanse USA Regulatory Address Cleanse
Address Type Summary report	ADDRTYPESUMMARY	Global Address Cleanse USA Regulatory Address Cleanse
Address Standardization Sample	ADDRINFOCODEDATA	Global Address Cleanse USA Regulatory Address Cleanse
Geocoder Summary Report (part of the US Addressing report)	GEO_ASSIGN_LEVEL GEO_INFO_CODE	Geocoder transform USA Regulatory Address Cleanse
Address Quality Code Summary	ADDRINFOCODESUMMARY	Global Address Cleanse
Best Record Summary	MTBRINFO MTBRACTION	Match
Match Contribution Report	MTRULESRES MTBRKGRPINFO MTBRKGRP	Match

Report	Repository table	Transform(s)
Match Criteria Summary report	MTCRITINFO	Match
	MTCRITDEF	
	MTKEYDEF	
	MTCMPCRIT	
Match Source Stats Summary report	MTGSSRCSTS	Match
	MTGSSRCBYSRCSTS	
Match Duplicate Sample report	MTDUPESDATA	Match
Match Input Source Output Select Report	MTINSRCINFO	Match
Match Multi-Source Frequency Report	Match statistics tables	Match
New Zealand SOA	SENDRIGHTADDRACCURACY	Global Address Cleanse
Data Cleanse Information Code Summary report	UDCINFOCODESUMMARY	Data Cleanse
Data Cleanse Status Code Summary report	UDCSTATUSCODESUMMARY	Data Cleanse

Related Information

[Data quality statistics tables and supplemental content information \[page 1344\]](#)

12.2.1 Repository tables for USA and Global address cleanse

The following table contains a list of repository tables used by the software to create postal regulatory and statistical reports for the USA Regulatory Address Cleanse and Global Address Cleanse transforms. The sections following this chart contain a topic for each table with descriptions for the fields (columns) in the table.



Repository table name	Description	Transform
ADDRINFOCODEDATA	Contains statistics about each record that generated an address information code during processing. Used for the Address Information Code Sample report.	Global Address Cleanse USA Regulatory Address Cleanse

Repository table name	Description	Transform
ADDRINFOCODESUMMARY	Contains statistics about each information code found during processing. Used for the Address Information Code Summary and the Address Quality Code Summary reports.	Global Address Cleanse USA Regulatory Address Cleanse
ADDRSTATUSCODE-DATA	Contains status codes found by the transform and their descriptions. Used for the Address Information Code Sample Report.	Global Address Cleanse USA Regulatory Address Cleanse
ADDRTYPESUMMARY	Contains statistics about address types found during processing. Used for the Address Type Summary.	Global Address Cleanse USA Regulatory Address Cleanse
ADDRVALIDATESUMMARY	Contains record validation statistics found by the transform. Used for the Address Validation Summary.	Global Address Cleanse USA Regulatory Address Cleanse
AMASADRPROCSUMMARY	Contains summary statistics for address processing. Used for the AMAS (Australia Matching Approval System) Address Matching Processing Summary.	Global Address Cleanse
CSLSTATS	Contains statistics from NCOALink processing. Used for the NCOALink CSL (Customer Service Log).	USA Regulatory Address Cleanse
DPVLACSLINKSUMMARY	Contains statistics about the DPV, DSF2, SuiteLink, and LACSLink processing. Used for the U.S. Addressing Report.	USA Regulatory Address Cleanse
DSFAUGMENTSTATS	Contains a detailed record plus the DSF2 licensee name and processing date. Used in the DSF2 Augment Statistics Log File.	USA Regulatory Address Cleanse
DSFSEQUENCESTATS	Contains DSF2 sequence statistics per postcode/sort-code combination. Used in the Delivery Sequence Invoice Report.	USA Regulatory Address Cleanse
MEDSTATS	Contains statistics for NCOALink move effective dates. Used for the NCOALink Report.	USA Regulatory Address Cleanse
NCOALCERTIFICATIONDATA	Contains statistics required for NCOALink processing. Used for the NCOALink Processing Summary Report.	USA Regulatory Address Cleanse

Repository table name	Description	Transform
NCOALINKSUMMARY	Contains summary statistics required for NCOALink processing. Used for the NCOALink Processing Summary Report.	USA Regulatory Address Cleanse
PAFBALASTATS	Contains statistics for NCOALink processing. Used for the Processing Acknowledgement Form and the Broker and List Administrator file.	USA Regulatory Address Cleanse
PSFORM3553DATA	Contains data for the USPS Form 3553 report that is submitted with all CASS-certified mailings.	USA Regulatory Address Cleanse
SERPADDRACCURACY	Contains statistics about the accuracy of the addresses in the list. Used for the Statement of Address Accuracy report (Canada).	Global Address Cleanse
SENDRIGHTADDRACCURACY	Contains summary statistics for New Zealand address processing. Used for the SendRight New Zealand Statement of Accuracy (SOA) report.	Global Address Cleanse
USREGULATORYLOCKING	Contains information about the record that caused the lock. Used for the US Regulatory Locking Report.	USA Regulatory Address Cleanse

12.2.1.1 ADDRINFOCODEDATA

This table contains statistics about each record that generated an address information code during processing. The information is used for the Address Information Code Sample report. It applies to the Global Address Cleanse and USA Regulatory Address Cleanse transforms.

Column	Data type definition	Description
 RECNUM	INT	Record number that generated the information code.
 NAME	VARCHAR (128)	Name of the field that generated the information code. For example, COUNTRY_CODE, ENGINE_NAME and so on.
VALUE	VARCHAR (256)	Information contained in the field identified under NAME. For example, "US" (COUNTRY_CODE) or "USA" (ENGINE_NAME).

i Note





The OBJECT_KEY column is included in this table but it is not listed. It is a primary key common to most tables. The OBJECT_ID column does not apply to this table.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.1.2 ADDRINFOCODESUMMARY

This table contains statistics about each information code found during processing. The information is used for the Address Information Code Summary report (USA Regulatory Address Cleanse transform) and the Address Quality Code Summary (Global Address Cleanse transform).

Column	Data type definition	Description
 COUNTRYID	VARCHAR (2)	Country code applicable to the listed INFOCODE.
 INFOCODE	VARCHAR (4)	Information code.
INFOCOUNT	INT	Total number of records in the list that received the listed INFOCODE.
 ENGINENAME	CHAR (50)	Name of the engine applicable to the listed INFOCODE. For the Global Address Cleanse transform, it is the name of the engine that processed the data (USA, CANADA, GLOBAL_ADDRESS, or GLOBAL_ADDRESS_CJK). For the USA Regulatory Address Cleanse transform, it is always USA.
 DATA_SOURCE_ID	VARCHAR (80)	Code that uniquely identifies the list that contains the listed INFOCODE.

i Note



The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.1.3 ADDRSTATUSCODEDATA

This table contains status codes and descriptions. The information is used for the Status Code Sample Report. It is applicable for Global Address Cleanse and USA Regulatory Address Cleanse transforms.

Column	Data type definition	Description
 RECNUM	INT	Record number that generated the information code.
 NAME	VARCHAR (128)	Name of the field that generated the information code. For example, COUNTRY_CODE, ENGINE_NAME.
VALUE	VARCHAR (256)	Information contained in the field identified under NAME. For example, "US" (COUNTRY_CODE) or "USA" (ENGINE_NAME).

Note



The OBJECT_KEY column is included in this table but it is not listed. It is a primary key common to most tables. The OBJECT_ID column is not applicable to this table.



Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.1.4 ADDRYPESUMMARY

This table contains statistics about address types found during processing for the Address Type Summary. The Address Type Summary contains the record count for each Assignment_Type field value (Global Address Cleanse transform) or Address_Type field value (USA Regulatory Address Cleanse transform).

Column	Data type definition	Description
 ENGINENAME	CHAR (50)	Name of the engine used to process the list. For the Global Address Cleanse transform, it is the name of the engine that processed the data (USA, CANADA, GLOBAL_ADDRESS, and GLOBAL_ADDRESS_CJK). For the USA Regulatory Address Cleanse transform, it is always USA.
 COUNTRYID	CHAR (2)	Country code applicable to the listed address type.
BLDNAMERECS	INT	Number of records that were determined to be building name addresses (Global Address Cleanse transform).
FIRMNAMERECS	INT	Number of records that were determined to be firm addresses.
GENDELIVERYRECS	INT	Number of records that were determined to be general delivery addresses.
UNIQSUBURBANRECS	INT	Number of records that were determined to be unique suburban addresses.

Column	Data type definition	Description
MOBILEROUTERECS	INT	Number of records that were determined to be mobile route addresses (Canadian addresses, Global Address Cleanse transform).
HIGHRISERECS	INT	Number of records that were determined to be highrise addresses.
MILITARYRECS	INT	Number of records that were determined to be military addresses.
POSTALRECS	INT	Number of records that were determined to be post office box addresses.
RURALRECS	INT	Number of records that were determined to be rural route addresses.
STREETRECS	INT	Number of records that were determined to be street addresses.
UNDERTERMINDRECS	INT, NULL	Number of records that were determined to be non US records (USA Regulatory Address Cleanse transform).
PROCESSEDRECS	INT	Number of records processed by the transform.
 SUITEL_TYPE	INT	Pre-SuiteLink or post-SuiteLink processing. Values include: 1 (Post-SuiteLink processing) -1 (Pre-SuiteLink processing)
 DATA_SOURCE_ID	NVARCHAR (80)	Unique identification code assigned to the list.

Note






The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.1.5 ADDRVALIDATESUMMARY

This table contains record validation statistics. The information is used for the Address Validation Summary. It is applicable for Global Address Cleanse and USA Regulatory Address Cleanse transforms.

Column	Data type definition	Description
 ENGINENAME	CHAR (50)	Name of the engine applicable to the COMPONENTNAME column. For the Global Address Cleanse transform, it is the name of the engine that processed the data (USA, CANADA, GLOBAL_ADDRESS, and GLOBAL_ADDRESS_CJK). For the USA Regulatory Address Cleanse transform, it is always USA.
 COUNTRYID	CHAR (2)	Country code applicable to the COMPONENTNAME column.
 COMPONENT-NAME	NVARCHAR (64)	Address component (ADDRESS1, LOCALITY1, and so on) to which the counts and percentages apply.
ADDEDRECS	INT	Number of records processed by the transform.
DELETEDRECS	INT	Number of records that were deleted during processing for the applicable COMPONENTNAME.
CORRECTEDRECS	INT	Number of records that were corrected during processing for the applicable COMPONENTNAME.
UNCHANGEDRECS	INT	Number of records that remained unchanged for the applicable COMPONENTNAME.
 SUITELINK_TYPE	INT	Indicates whether the count information applies to pre-SuiteLink or post-SuiteLink processing, or pre-NCOALink or post-NCOALink processing. Values include: 1 (post-SuiteLink/NCOALink processing) -1 (pre-SuiteLink/NCOALink processing)
 DATA_SOURCE_ID	NVARCHAR (80)	Unique identification code assigned to the list.

i Note




The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.1.6 CSLSTATS

This table contains statistics for NCOALink processing for the Customer Service Log. It is applicable for the USA Regulatory Address Cleanse transform.

Column	Data type definition	Description
 LICENSEEID	NVARCHAR (4)	NCOALink licensee's identification number assigned by the USPS.
 SEQNUM	INT	DSF2 sequence number assigned to the address.
DTLRECORD	NVARCHAR (1000)	String of Customer Service Log field values.
LICENSEETYPE	NVARCHAR (1)	Type of service provider. Values are: E (End user) F (Full service provider) L (Limited service provider)
PROCESSINGDATE	DATETIME	Date and time the job was processed through the software in YYYY-MM-DD HR:MIN:SEC format.
RESERVED_COUNT1	INT	Reserved for future record counts.
RESERVED_COUNT2	INT	Reserved for future record counts.
RESERVED_COUNT3	INT	Reserved for future record counts.
RESERVED_STRING1	NVARCHAR (100)	Reserved for future strings.
RESERVED_STRING2	NVARCHAR (100)	Reserved for future strings.
RESERVED_STRING3	NVARCHAR (100)	Reserved for future strings.
 DATA_SOURCE_ID	NVARCHAR (80)	Unique identification code assigned to the list.

i Note


The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.


Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.1.7 DPVLACSLINKSUMMARY

This table contains statistics about the DPV, DSF2, LACSLink, and SuiteLink processing. The information is used in the US Addressing Report. It is applicable for the USA Regulatory Address Cleanse transform.

Column	Data type definition	Description
LACSCONVERTED_A	INT	Number of LACSLink records that were converted by the transform (return code A).
LACSCONVERTED_92	INT	Number of LACSLink records that matched after dropping the secondary number from the input address (return code 92).
LACSNOTCONVERTED_09	INT	Number of LACSLink records that matched an input address to an old address, and the old address is a high-rise default address. No new addresses provided (return code 09).
LACSNOTCONVERTED_00	INT	Number of records that had no match to LACSLink and, therefore, no addresses converted (return code 00).
LACSNOTCONVERTED_14	INT	Number of LACSLink records that matched to LACSLink, but couldn't be converted to a deliverable address (return code 14).
DPVVALIDATED_Y	INT	Number of records that were validated for DPV (status of Y, primary and secondary range is valid).
DPVVALIDATED_S	INT	Number of records that contained an invalid secondary range (DPV status of S, secondary range not present).
DPVVALIDATED_D	INT	Number of records that did not have a secondary range (DPV status of D, secondary range not present).
DPVVALIDATED_CMRA	INT	Number of records that were validated as a CMRA (Commercial Mail Receiving Agency).
DPVNOTVALIDATED	INT	Number of records that were not validated for DPV.
 SUITEL_TYPE	INT	Indicates whether the information is from pre-SuiteLink or post-SuiteLink processing. 1 (Pre-SuiteLink processing) -1 (Post-SuiteLink processing) 0 (NCOALink is not enabled so there is no pre-NCOALink or post-NCOALink processing sections)
SUITEL_MATCH_A	INT	Number of records that matched to SuiteLink and had secondary information added.
SUITEL_NOMATCH_00	INT	Number of records that did not match to SuiteLink.
DPVVALIDATED_VACANT	INT	Number of records with DPV status of vacant.
DPVVALIDATE_NOSTATS	INT	Number of records with DPV status of nostats.
DSF2_DROP	INT	Number of records that are dropped at a delivery point that serves businesses or families (for example, a CMRA).
DSF2_BUSINESS	INT	Number of records that have a business address.
DSF2_THROWBACK	INT	Number of records that are throwbacks (customer wants delivery at PO Box instead of street address).
DSF2_SEASONAL	INT	Number of records that are seasonally occupied.
DSF2_EDUCATIONAL	INT	Number of records that are educational institutions.
DSF2_CURB	INT	Number of records that have a curb-side delivery indicator.

Column	Data type definition	Description
DSF2_CENTRAL	INT	Number of records that have a central delivery indicator.
DSF2_DOOR	INT	Number of records that have a door-slot delivery indicator.
DSF2_NDCBU	INT	Number of records that have an NDCBU (Neighborhood Delivery Centralized Box Unit) delivery indicator.
 DATA_SOURCE_ID	NVARCHAR (80)	A unique identification for a list.

Note




The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.1.8 DSFAUGMENTSTATS

This table contains information used in the DSF2 Augment Statistics Log file. It is applicable for the USA Regulatory Address Cleanse transform.

Column	Data type definition	Description
 LICENSEE_NUMBER	NVARCHAR (4)	DSF2 licensee number.
 SEQNUM	INT	Sequence number from 0000 to 9999. <ul style="list-style-type: none"> If the Postcode2 field is blank, this column is blank. If the Postcode2 field is not blank and the DPV_Status is not Y, then the column contains "0000".
LICENSEE_NAME	NVARCHAR (40)	DSF2 licensee name.
PROCESSING_DATE	DATETIME	Date and time the job was processed through the software in YYYY-MM-DD HR:MIN:SEC format.
DTLRECORD	NVARCHAR (1000)	String of log file field values for DSF2 Augment Statistics.
 DATA_SOURCE_ID	NVARCHAR (80)	Unique identification code assigned to the list.

Note




The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.1.9 DSFSEQUENCESTATS

This table contains information for the Delivery Sequence Invoice Report. It is applicable for the USA Regulatory Address Cleanse transforms.

Column	Data type definition	Description
 POSTCODE1	NVARCHAR (5)	Postcode (ZIP Code) for the records being processed.
 SORTCODE_ROUTE	NVARCHAR (4)	Sortcode route (carrier route) for the records being processed.
 LICENSEE_NAME	NVARCHAR (40)	DSF2 licensee name.
TOTAL_DELIVERY_POINTS	INT	Number of deliveries within the specified postcode/sortcode combination.
TOTAL_RESIDENCES	INT	Number of residences within the specified postcode/sortcode combination.
TOTAL_DELIVERY_POINTS_SEQ	INT	Number of delivery points sequenced by the transform for the specific postcode/sortcode combination.
TOTAL_RESIDENCES_SEQ	INT	Number of residences sequenced for the specific postcode/sortcode combination.
SITE_LOCATION	NVARCHAR (20)	Location of the site where the DSF2 walk sequence processing occurred.
LIST_ID	NVARCHAR (6)	Unique ID assigned by the licensee to identify the list.
PROCESSING_DATE	DATETIME	Date and time the job was processed through the software in YYYY-MM-DD HR:MIN:SEC format.

Note




The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.1.10 MEDSTATS

This table contains statistics for NCOALink move effective dates. The information is used for the NCOALink Report. It is applicable for the USA Regulatory Address Cleanse transform.

Column	Data type definition	Description
 CATEGORY	NVARCHAR (1)	Return code category: A = Return codes A, 91, 92 B = Return codes 01, 02, 03 C - Return codes 05, 14, 19
 PERIOD	NVARCHAR (2)	Time period for the number of records for each return code category. Values include the following: 1 = Months 0-3 2 = Months 4-6 3 = Months 7-12 4 = Months 13-18 5 = Months 19+
MEDCOUNT	INT	Number of moves in the specified period.
 DATA_SOURCE_ID	NVARCHAR (80)	Unique identification for a list.

i Note

The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information


[Repository tables common columns \[page 1297\]](#)

12.2.1.11 NCOALCERTIFICATIONDATA

This table contains statistics from NCOALink processing and is used for the NCOALink Reports. It is applicable for the USA Regulatory Address Cleanse transform.

Column	Data type definition	Description
MAILER_PAF_ID	VARCHAR (40)	Combination of the following information: <ul style="list-style-type: none"> • USPS-assigned license ID (first 4 characters) • List Owner NAICS Code (next 5 characters) • Frequency of processing (next 2 characters) • Licensee-assigned List ID (last 6 characters)
MAILER_COMPANY	VARCHAR (30)	Name of the customer that requested NCOALink processing.
PROCESSES_USED	VARCHAR (768)	All USPS processes used to obtain the final data results.
DATA_RETURNED	VARCHAR (1)	Purpose for NCOALink processing. C (Change of address) F (Return codes) S (Statistics)
PRE_PROCESSES	VARCHAR (1)	Indicates that the list was pre processed and what data modifications occurred. Values are: N (None) Y (Yes, but with no data modifications). D (Yes, data modifications from sources other than postal data). P (Yes, data modifications from postal data only. For example, ZIP+4 or DPV) B (Yes, data modifications from postal and other sources)
POST_PROCESSES	VARCHAR (1)	Indicates that the list was post processed and what data modifications occurred. Values are: N (None). Y (Yes, but with no data modifications). D (Yes, data modifications from sources other than postal data). P (Yes, data modifications from postal data only. For example, LACSLink). B (Yes, data modifications from postal and other sources).
CONCUR- RENT_PROCESSES	VARCHAR (1)	Indicates that the list was concurrently processed and the data modifications that occurred. Values are: N (None). Y (Yes, but with no data modifications). D (Yes, data modifications from sources other than postal data). P (Yes, data modifications from postal data only. For example, ZIP+4 or DPV). B (Yes, data modifications from postal and other sources).
LIST_NAME	VARCHAR (30)	Name of the list being processed.

Column	Data type definition	Description												
LICENSEE_ID	VARCHAR (4)	Unique ID for the broker or list administrator.												
LICENSEE_COMPANY	VARCHAR (50)	Broker or list administrator name.												
PROCESSING_CATE- GORY	VARCHAR (20)	Reason for the NCOALink processing. Values are: <table><tr><th>Option</th><th>Printed on report</th></tr><tr><td>Marketing</td><td>MKTG TEST</td></tr><tr><td>Normal</td><td>NORMAL</td></tr><tr><td>Stage I</td><td>STAGE I</td></tr><tr><td>Stage II</td><td>STAGE II</td></tr><tr><td>System testing</td><td>SYS TEST</td></tr></table>	Option	Printed on report	Marketing	MKTG TEST	Normal	NORMAL	Stage I	STAGE I	Stage II	STAGE II	System testing	SYS TEST
Option	Printed on report													
Marketing	MKTG TEST													
Normal	NORMAL													
Stage I	STAGE I													
Stage II	STAGE II													
System testing	SYS TEST													
SERVICE_PROVIDER	VARCHAR (1)	NCOALink list processor's provider level. Values are: 0 End user 1 Limited service provider 2 Full service provider												
NCOAL_PROC- ESS_DATE	VARCHAR (10)	Date based on the time that the list was processed. Automatically gener- ated by the software, and included in reports.												
LIST_RETURN_DATE	VARCHAR (10)	Date obtained from the <i>List returned date</i> option in the USPS License In- formation group.												
DPV_ENABLED	VARCHAR (1)	DPV is enabled in the transform (Y/N indicator in reports).												
LACSL_ENABLED	VARCHAR (1)	LACSLink is enabled in the transform (Y/N indicator in reports).												
SUITEL_ENABLED	VARCHAR (1)	SuiteLink is enabled in the transform (Y/N indicator in reports).												
ANK_ENABLED	VARCHAR (1)	ANKLink is enabled in the transform (Y/N indicator in reports).												
USE_BUSI- NESS_MOVES	VARCHAR (1)	Business moves were included in processing (Y/N indicator in reports).												
USE_INDIVID- UAL_MOVES	VARCHAR (1)	Individual moves were included in processing (Y/N indicator in reports).												
USE_FAMILY_MOVES	VARCHAR (1)	Family moves were included in processing (Y/N indicator in reports).												
MAIL_CLASS	VARCHAR (1)	Mail class that was processed.												
MATCH_LOGIC	VARCHAR (1)	Move types that were processed. Values are: S (Standard move types such as business, individual, and family matches). I (Individual only). B (Business only). C (Individual and business only).												

Column	Data type definition	Description
STD_OUTPUT_RE- TURNED	VARCHAR (1)	<p>Transform-generated value that indicates whether the standard output was returned. Values are:</p> <p>Y (All NCOALink-required output returned to client).</p> <p>N (Post processes modified return information. For example updates applied to list).</p> <p>B (Post processes modified return information. However, a separate file containing all required output data was also returned).</p>
PROCESS_PERIODI- CAL	VARCHAR (1)	Periodicals mail was processed by the transform. (Y/N indicator in reports.)
PROCESS_FIRST	VARCHAR (1)	First Class mail was processed by the transform. (Y/N indicator in reports.)
PROCESS_STD	VARCHAR (1)	Standard mail was processed by the transform. (Y/N indicator in reports.)
PROCESS_PACKAGE	VARCHAR (1)	Package Services mail was processed by the transform. (Y/N indicator in reports.)
TOTAL_RECORDS	INT	Number of records processed by the transform.
NCOAL_MATCHES	INT	Number of NCOALink matches found in the list.
ANK_MATCHES	INT	Number of ANKLink matches found in the list.
ZIP4_MATCHES	INT	Number of ZIP+4 matches found in the list.
DPV_MATCHES	INT	Number of DPV matches found in the list.
LACSL_MATCHES	INT	Number of LACSLink matches found in the list.
SUITEL_MATCHES	INT	Number of SuiteLink matches found in the list.
INDIVIDUAL_RE- CORDS	INT	Number of individual type records found in the list.
FAMILY_RECORDS	INT	Number of family type records found in the list.
BUSINESS_RECORDS	INT	Number of business type records found in the list.
NCOAL_NAME	VARCHAR (16)	Name of the NCOALink-certified software.
NCOAL_VERSION	VARCHAR (16)	Version of the NCOALink-certified software.
RESERVED_COUNT1	INT	Extra field for additional counts.
RESERVED_COUNT2	INT	Extra field for additional counts.
RESERVED_COUNT3	INT	Extra field for additional counts.
RESERVED_STRING1	VARCHAR (100)	Extra field for additional string.
RESERVED_STRING2	VARCHAR (100)	Extra field for additional string.
RESERVED_STRING3	VARCHAR (100)	Extra field for additional string.
 DATA_SOURCE_ID	VARCHAR (80)	Unique identification code assigned to the list.

i Note



The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.1.12 NCOALINKSUMMARY

This table contains summary statistics required for NCOALink processing. The information is used for the NCOALink Summary Report. It is applicable for the USA Regulatory Address Cleanse transform.

Column	Data type definition	Description
 CODE	NVARCHAR (5)	Return code.
CODEGROUP	NVARCHAR (1)	Group in which the code applies. 1 = New address provided by NCOALink 2 = Found COA 3 = Cannot match COA 4 = From the daily delete process
CODEDESC	NVARCHAR (50)	Description for the return code listed.
RECCOUNT	INT	Number of list records found for the code listed.
ADDRESS_PROVIDED	NVARCHAR (1)	Indicates whether an address was provided (Y/N).
ADDRESS_SOURCE	NVARCHAR (1)	D = Derived by data S = Derived from software
DETAILED_DESC	VARCHAR (1)	Provides a more detailed description of the listed code.
 DATA_SOURCE_ID	NVARCHAR (80)	Unique identification code assigned to a list.

i Note





The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.1.13 PAFBALASTATS

This table contains statistics for NCOALink processing. The information is used for the Processing Acknowledgement Form and the Broker and List Administrator file. It is applicable for the USA Regulatory Address Cleanse transform.

Column	Data type definition	Description
 LICENSEEID	NVARCHAR (4)	Service provider's NCOALink license ID.
DTLRECORD	NVARCHAR (700)	All field values per list.
 LOGTYPE	NVARCHAR (1)	The type of log file: P = PAF B = BALA
 SEQNUM	INT	DSF2 sequence number assigned to the address.
RESERVED_COUNT1	INT	Reserved for future record counts.
RESERVED_COUNT2	INT	Reserved for future record counts.
RESERVED_COUNT3	INT	Reserved for future record counts.
RESERVED_STRING1	NVARCHAR (255)	Reserved for future text.
RESERVED_STRING2	NVARCHAR (255)	Reserved for future text.
RESERVED_STRING3	NVARCHAR (255)	Reserved for future text.
 DATA_SOURCE_ID	NVARCHAR (80)	Unique identification for a list.

i Note

The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.


Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.1.14 PSFORM3553DATA

This table contains data for the USPS Form 3553 report that is submitted with all CASS-certified mailings.

Column	Data type definition	Description
CASSCOMPANYNAME	NVARCHAR (50)	Name of the company that is CASS-certified.
CASSCONFIGURATION	NVARCHAR (10)	Software configuration settings as they appear on the CASS certificate.
CASSSOFTWARE	NVARCHAR (60)	Name and version of the software that is CASS certified.
Z4COMPANYNAME	NVARCHAR (40)	Name of the company that is Z4Change certified.
Z4CONFIGURATION	NVARCHAR (10)	Software configuration settings for Z4Change.
Z4SOFTWARE	NVARCHAR (60)	Name and version of the software that is certified for Z4Change processing.
eLOTCOMPANYNAME	NVARCHAR (40)	Name of the company that is certified for eLOT processing.
eLOTCONFIGURATION	NVARCHAR (10)	Software configuration settings for eLOT.
eLOTSOFTWARE	NVARCHAR (60)	Name and version of the software that is certified for eLOT processing.
MASSCOMPANYNAME	NVARCHAR (40)	Not populated from the software.
MASSCONFIGURATION	NVARCHAR (10)	Not populated from the software.
MASSSOFTWARE	NVARCHAR (60)	Not populated from the software.
MLOCRSERIALNUMBER	NVARCHAR (20)	Not populated from the software.
LISTPROCESSOR	NVARCHAR (35)	MASS list processor. Not populated from the software.
ZIP4DBDATE	VARCHAR (15)	ZIP4 directory date.
Z4CHANGEDATE	VARCHAR (15)	Not populated from the software.
Z4CHANGEDBDATE	VARCHAR (15)	Z4Change directory date.
eLOTDATE	VARCHAR (15)	Not populated from the software.
eLOTDBDATE	VARCHAR (15)	eLOT directory date.
CRISDATE	VARCHAR (15)	Not populated from the software.
CRISDBDATE	VARCHAR (15)	Not populated from the software.
ADDRESSLISTNAME	NVARCHAR (20)	Name of the address list processed. (Entered in CASS Report Options  List Name )
NUMBEROFLISTS	INT	Not populated from the software.
TOTALRECPROCESSED	INT	Number of records processed in the job.
ZIP4RECORDSCODED	INT	Number of ZIP+4 records that were DPV confirmed.
Z4CHGRECORDSCODED	INT	Number of records that were Z4Change coded.
DPBRECORDSCODED	INT	Number of records with delivery point bar codes.
DPBCDATEFROM	VARCHAR (15)	Not populated from the software.

Column	Data type definition	Description
FIVEDIGITRECCODED	INT	Number of records that were assigned 5-digit ZIP Codes.
FIVEDIGITDATEFROM	VARCHAR (15)	Not populated from the software.
CRRTRECORDSCODED	INT	Number of records that were assigned carrier route codes.
CRRDATEFROM	VARCHAR (15)	Not populated from the software.
eLOTRECORDSCODED	INT	Number of records that were assigned eLOT codes.
eLOTDATEFROM	VARCHAR (15)	Not populated from the software.
MAILERINFO1	NVARCHAR (50)	Information entered in the USA Regulatory Address Cleanse transform in the CASS report options group.
MAILERINFO2	NVARCHAR (50)	Information entered in the USA Regulatory Address Cleanse transform in the CASS report options group.
MAILERINFO3	NVARCHAR (50)	Information entered in the USA Regulatory Address Cleanse transform in the CASS report options group.
MAILERINFO4	NVARCHAR (50)	Information entered in the USA Regulatory Address Cleanse transform in the CASS report options group.
HIGHRISEDEFAULT	INT	Number of records that were assigned as highrise defaults.
HIGHRISEEXACT	INT	Number of records that were assigned as highrise exact matches.
RURALROUTEDEFAULT	INT	Number of records that were assigned as rural route default matches.
RURALROUTEEXACT	INT	Number of records that were assigned as rural route exact matches.
LACS	INT	Number of addresses that were converted through the LACSLink process.
EWS	INT	Number of records that were assigned as EWS addresses (and, therefore, are not listed in the current U.S. Postal Service® ZIP + 4 File).
DPV	INT	Number of records that were confirmed as ZIP + 4/DPV that matched to a highrise default, and the SuiteLink process returned the appropriate suite number.
RDI	INT	Not populated from the software (always zero).
RESERVED_COUNT1	INT	Reserved for future record counts.
RESERVED_COUNT2	INT	Reserved for future record counts.
RESERVED_COUNT3	INT	Reserved for future record counts.
RESERVED_STRING1	NVARCHAR (100)	Reserved for future strings.
RESERVED_STRING2	NVARCHAR (100)	Reserved for future strings.
RESERVED_STRING3	NVARCHAR (100)	Reserved for future strings.
 DATA_SOURCE_ID	NVARCHAR (80)	Unique identification assigned to the list.

Note

The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.1.15 SENDRIGHTADDRACCURACY

This table contains summary statistics about the address processing for the Global Address Cleanse SendRight certification report. The information is used for the SendRight Address Accuracy report (for New Zealand). These statistics are generated by the Global Address Cleanse transform.

Column	Data type definition	Description
CERTCOMPANYNAME	VARCHAR (40)	Name of the company that is SendRight certified. This value will be SAP Business Objects.
CERTPRODNAME	VARCHAR (40)	Name of the product that is SendRight certified. This value will be Data Services.
CERTPRODVER	VARCHAR (20)	Version of the SendRight-certified software.
PAFVERSION	VARCHAR (20)	Version of the current (most recent) PAF. PAF refers to the version number supplied by New Zealand post.
SOAUNIQUEID	VARCHAR (20)	Unique ID generated by the certified engine to guarantee a unique report for each Global Address Cleanse transform within a data flow.
CUSTOMERNAME	VARCHAR (40)	Name of the list owner or list processor.
MAILERADDR1–6	VARCHAR (60)	Name and address of the person or organization for whom you are preparing the mailing. (6 columns, 60 varchars each).
FILENAME	VARCHAR (40)	Input file name.
TOTALRECPROC	INT	Number of records processed.
UNIQUEMATCHES	INT	Number of unambiguous matches between the input addresses and one address record in the PAF.
BASEMATCHES	INT	Number of unambiguous matches between the input addresses and one base address.
DATEISSUED	DATETIME	Date that this report was generated.
DATEEXPIRED	DATETIME	Date that this report will expire. (Always exactly 1 year after generation date.)
TOTALVALIDATED	INT	Number of records validated.
ADDRESSACCURACY	VARCHAR (8)	Percentage of the total records validated from the total number of records processed.
RESERVED_COUNT1–3	INT	Reserved for future use.
RESERVED_STRING1–3	VARCHAR (255)	Reserved for future use.

12.2.1.16 SERPADDRACCURACY

This table contains statistics about the accuracy of the addresses in the list. The information is used for the SERP (Statement of Address Accuracy) report (Canada). These statistics are generated by the Global Address Cleanse transform.

Column	Data type definition	Description
VENDORNAME	VARCHAR (40)	Name of the Address Accuracy vendor. This value will be SAP Business Objects.
VENDORADDR1-2	VARCHAR (40)	Address of the Address Accuracy vendor. This will be the current SAP Business Objects address. (Two columns, 40 varchars each.)
PRODNAMEVER	VARCHAR (36)	Name of the Address Accuracy software product and version. This is hardcoded to the current product and version.
SREXPYDATE	VARCHAR (11)	Expiration date of the software's SERP certification.
CUSTOMERNAME	VARCHAR (40)	Company name of the organization for whom the mailing is being prepared.
CUSTOMERADDR1-4	VARCHAR (40)	Name and address of the person or organization for whom the mailing is being prepared. (Four columns, 40 varchars each.)
CUSTOMERCPNUM	VARCHAR (15)	Date of the postalcode file.
CPCMASTERFILEVER	VARCHAR (11)	Customer's CPC number that is located in the Canada Post Contract.
TOTALRECPROC	INT	Number of urban and rural records processed.
RECASSIGNED	INT	Number of records assigned.
QUEST_RECS_RURAL	INT	Number of questionable rural records.
QUEST_RECS_APPT	INT	Number of questionable apartment records.
RESERVED_COUNT1-3	INT	Reserved for future record counts.
RESERVED_STRING1-3	NVARCHAR (255)	Reserved for future strings.

i Note



The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.1.17 USREGULATORYLOCKING

This table contains information about the record that caused DPV or LACSLink locking. The information is used for the USA Regulatory Locking Report.

Column	Data type definition	Description
 LOCKMODE	INT	Whether the lock was for DPV (1) or LACSLink (2).
LOCK_CODE	VARCHAR (81)	Lock code issued by the software and found in the Alert area of the report.
POSTCODE	VARCHAR (81)	Input postcode from the address that caused the lock.
PRIMARY_NUMBER	VARCHAR (81)	Input primary range from the address that caused the lock.
PRIMARY_NAME	VARCHAR (81)	Primary name from the address that caused the lock.
UNIT_NUMBER	VARCHAR (81)	Secondary range for the address that caused the lock.
UNIT_DESCRIPTION	VARCHAR (81)	Unit designator for the address that caused the lock.
PRIMARY_PREFIX	VARCHAR (81)	Predirectional for the address that caused the lock.
ADDRESS_TYPE	VARCHAR (81)	Suffix for the address that caused the lock.
PRIMARY_POSTFIX	VARCHAR (81)	Postdirectional for the address that caused the lock.
RESERVED_COUNT1-3	INT	Reserved for future record counts.
RESERVED_STRING1-3	NVARCHAR (255)	Reserved for future strings.
 DATA_SOURCE_ID	NVARCHAR (80)	Unique identifier for the list.

i Note

The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.2 Repository tables and reports for Data Cleanse

The following table contains a list of repository tables used for statistical reports related to status and information codes generated from the Data Cleanse transform.

Repository table name	Description	Transform
UDCSTATUSCODESUMMARY	<p>Lists status codes that were generated during a specific data cleanse job and the total number of records assigned to that code.</p> <p>Used for the Data Cleanse Status Code Summary report</p>	Data Cleanse
UDCINFOCODESUMMARY	<p>Lists information codes that were generated during a specific data cleanse job and the total number of records assigned to that code.</p> <p>Used for the Data Cleanse Information Code Summary</p>	Data Cleanse

Related Information

[Status codes \(Data Cleanse\) \[page 824\]](#)


[Information codes \(Data Cleanse\) \[page 827\]](#)

[Data quality statistics tables and supplemental content information \[page 1344\]](#)

12.2.2.1 UDCINFOCODESUMMARY

A Data Cleanse repository table that contains statistics about the information codes that were generated by a specific data cleanse job.

The software uses the information from this table for the Data Cleanse Information Code Summary report.

Column	Data type definition	Description
 INFOCODE	VARCHAR (4)	The information code. Each code begins with a letter that indicates the parser followed by three numbers that represent specific classifications.

Column	Data type definition	Description
INFOCOUNT	INT	The number of records that were assigned this information code.
PARSER	NVARCHAR (100)	The parser used to determine the status. Parsers are: <ul style="list-style-type: none"> • Custom • Date • Firm • Person • Phone • SSN
PARSE_NUMBER	INT	A number that corresponds to the output field when there is more than one instance. For example Person 1 has a parse number 1, Person 2 has a parse number 2, and so on.
RESERVED_COUNT1-3	INT	Reserved for future use.
RESERVED_STRING1-3	NVARCHAR (255)	Reserved for future use.

i Note

This table also contains the common columns OBJECT_KEY and OBJECT_ID. The primary key is made up of the OBJECT_KEY, OBJECT_ID, PARSER, and the INFOCODE.

Related Information


[Repository tables common columns \[page 1297\]](#)

[Information codes \(Data Cleanse\) \[page 827\]](#)

12.2.2.2 UDCSTATUSCODESUMMARY

A Data Cleanse repository table that contains information about the status codes that were generated from a specific data cleanse job.

The software uses the information from this status table for the Data Cleanse Status Code Summary report. Data Cleanse status codes indicate how the data was standardized, and that a specific standard could be used on the data.

Column	Data type definition	Description
 STATUSCODE	NVARCHAR (100)	The status code. The status code format is <code><parser name>_<output field name>_STD</code> .
STATUSCOUNT	INT	The number of records that were assigned this status code.
PARSER	NVARCHAR (100)	The parser used to determine the status. Parsers are: <ul style="list-style-type: none"> • Custom • Date • Firm • Person • Phone • SSN
OUTPUT_FIELD	NVARCHAR (88), NULL	The output field name.
RESERVED_COUNT1-3	INT	Reserved for future use.
RESERVED_STRING1-3	NVARCHAR (255)	Reserved for future use.

i Note

This table also contains the common columns OBJECT_KEY and OBJECT_ID. The primary key is made up of the OBJECT_KEY, OBJECT_ID, and the STATUSCODE.

Related Information

[Repository tables common columns \[page 1297\]](#)

[Status codes \(Data Cleanse\) \[page 824\]](#)

12.2.3 Repository tables and reports for Geocoder

The Geocoder transform generates statistics that the software uses in US postal reporting.

The following table contains a list of repository tables with statistics used for the Geocoder Summary report and the US Addressing report. The sections following this chart contain a topic for each table with descriptions for the fields (columns) in the table.



Repository table name	Description	Transform
GEO_ASSIGN_LEVEL	Contains Geo statistics per assignment level. Used in the Geocoder Summary Report, a subsection of the US Addressing Report.	Geocoder
GEO_INFO_CODE	Contains Geocoder information codes and record counts per code. Used in the US Addressing Report.	Geocoder

Related Information

[Data quality statistics tables and supplemental content information \[page 1344\]](#)

12.2.3.1 GEO_ASSIGN_LEVEL

Statistics generated in this repository table are found in the Geocoder Summary Report. The Geocoder transform is used in conjunction with the Global Address Cleanse transform or the USA Regulatory Address Cleanse transform.

Column	Data type definition	Description
 COUNTRY	CHAR (2)	Three-character ISO country code.
 CODE	NVARCHAR (4)	Code that provides information about the geocoding results. Values are: PRE (Primary number exact) PRI (Primary number interpolated) PF (Postcode full) P2P (Postcode2 partial) P1 (Postcode1) L4 (Locality4) L3 (Locality3) L2 (Locality2) L1 (Locality1)
ASSIGN_COUNT	INT	Number of records for the assignment level listed.

i Note

The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.


Related Information

[Reference Guide: Geocoder fields, Output fields \[page 539\]](#)

[Repository tables common columns \[page 1297\]](#)

12.2.3.2 GEO_INFO_CODE

The following table contains statistics about information code found during processing for the Geocoder transform. The Geocoder transform that is used in conjunction with the Global Address Cleanse transform or the USA Regulatory Address Cleanse transform.

Column	Data type definition	Description
 INFOCODE	NVARCHAR (4)	A three-character code that provides information about the geocoding results. The status for address and point-of-interest geocoding assignment is indicated in the third character. The status for reverse geocoding assignment is indicated in the second and third characters. If assigned to the best level, the Info_Code field is blank. The first character is not used at this time.
INFO_COUNT	INT	Number of records for the listed INFOCODE.

Note

The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Output fields for the Geocoder transform \[page 539\]](#)

[Repository tables common columns \[page 1297\]](#)

12.2.4 Match repository statistics tables

There are many Match statistics that appear in various reports. These statistics are stored in statistics tables in the repository. The Match repository statistics tables are listed alphabetically below with a brief description for each table. Later, each table is described including the field names, data-type definitions, and descriptions.

Table	Description
MTBRACTION	Information about the best record action. Related report: Best Record Summary report.
MTBRINFO	Best record information. Related report: Best Record Summary report.



Table	Description
MTBRKGRP	Break group information. Related report: Break Group Contribution section of the Match Contribution report.
MTBRKGRPINFO	Information about the top and bottom ten break group strings and counts. Related report: Match Contribution report.
MTCMPCRIT	Match level option setting information for each match set. Related report: Match Level Options section of the Match Criteria Summary report.
MTCRITDEF	Match criteria information for each match set. Related report: Match Input Fields and Detailed Criteria Definition sections of the Match Criteria Summary report.
MTDUPESDATA	Information about match records as a sample of the match results. Related report: Duplicate Sample report.
MTGSSRCBYSRCSTS	Inter and intra-source match counts. Related report: Match Source Statistics Summary report.
MTGSSRCSTS	Information about the distribution of the matches found in various input source records including how the matches were distributed as master records and subordinate records. Related report: Match Source Statistics report.
MTINFO	Name of the match set. Related reports: All of the Match reports except the Best Record Summary report.
MTINSRCBYSRC	Information about how often each input source matched the other input sources.
MTINSRCGRPINFO	Input source group information. The table is populated once per transform if statistics are enabled and the source groups section of the input source object is defined.
MTINSRCINFO	Input source information. The table is populated once per transform if statistics are enabled and the input source object is defined.
MTINSRCMSRC	Multi-source statistics of each input source and each source group. The table is populated once per input source group statistics object if statistics are enabled.
MTINSRCSELECT	Input source group information. The table is populated once per input source select record object if statistics are enabled.
MTINSRCSTATS	Statistics of each input source and each source group. The table is populated once per input source group statistics object if statistics are enabled.
MTKEYDEF	Preprocessing criteria information for each key field. Related report: Match Input Fields section of the Match Criteria Summary report.
MTPROCESS	Information about match processing. Related reports: All Match reports except the Best Record Summary report.
MTRULESRES	Information about the effect of the criteria on the total matching process. Related report: Match Contribution report (Criteria Information sub report).

12.2.4.1 MTBRACTION

This table contains best record information and is applicable to the Match transform. The information is used for the Best Record Summary report.

i Note

This table is populated only if the Best Record functionality is enabled in your job.

Column	Data type definition	Description
 PROCID	INT	Sequential number that identifies a match level or an association.
 ACTIONID	INT	Sequential number that identifies a Best Record Action section.
BRNAME	NVARCHAR (15)	Name of the Best Record operation you specified in the Match Editor.
SRCFLD	NVCHAR (256)	Source field used in the Best Record Action section.
DSTFLD	NVCHAR (256)	Destination field used in the Best Record Action section.

i Note

This column is blank if a source expression is completed instead of a source field.

i Note

The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information



[Repository tables common columns \[page 1297\]](#)

12.2.4.2 MTBRINFO

This table contains best record information and is applicable to the Match transform. The information is used for the Best Record Summary report.

i Note

This table is populated only if the Best Record functionality is enabled in your job.

Column	Data Type Definition	Description
 PROCID	INT	Sequential number that identifies a match level or an association.
 BRNAME	NVARCHAR (255)	Name of the best record object.
POSTDEST	CHAR (1)	Destination for the post. Values are A (all), M (master only), or S (subs only).
POSTOPD	CHAR (1)	More than one posting per destination will be attempted or not for each record (Y/N).

Column	Data Type Definition	Description
PROTECTDROPS	INT	Number of post operations that were canceled because the destination record was protected.
DSTFLDDROPS	INT	Number of post operations that were canceled because the destination record had already been posted to once, and POSTOPD was set to Y.
FILTERDROPS	INT	Number of post operations that were canceled because the Best Record filter returned F (false).
POSTCOMPLETS	INT	Number of post operations that were successfully completed.

i Note

The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.4.3 MTBRKGRP

This table contains information about match break groups and is applicable to the Match transform. The information is used in the Match Contribution Report. This table is generated if matching is performed (not associating).

i Note

If breaking is not defined in the job setup, then all records are included in one break group.

Column	Data type definition	Description
NUMRECS	INT	Number of records processed in all of the break groups.
ELAPSEDTIME	INT	Elapsed time to process the break groups (in seconds).
CMPBUFMAXRECS	INT	Maximum number of records that can fit into the compare buffer at one time.
NOBRKCOMPARES	FLOAT	Number of comparisons that would be made without using any break group strategy (or putting all records in a single break group).
BRKGRPCOUNT	INT	Number of break groups formed based on the break group strategy.
BRKGRPLARGEST	INT	Largest break group processed.
BRKGRPCOMPARES	FLOAT	Number of comparisons made in all the break groups.

i Note

The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information


[Repository tables common columns \[page 1297\]](#)

12.2.4.4 MTBRKGRPINFO

This table contains information about the top and bottom ten break group strings and counts and is applicable to the Match transform. The information is used in the Break Group Contribution section of the Match Contribution Report.

i Note

This table is populated only if the software performs matching and breaking.

Column	Data type definition	Description
 BRKID	INT	Identification number for the break group.
BRKSTR	NVARCHAR (256)	Break string from the break group.
NUMRECS	INT	Number of records in the break group.

i Note


The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.4.5 MTCMPCRIT

This table contains match level option setting information for each match set and is applicable to the Match transform. This information is in the Match Level Options section of the Match Criteria Summary report.

Column	Data type definition	Description
 CMPCCRITID	INT	Sequential number that identifies the criteria of a match level.
NAME	NVARCHAR (256)	Match level name.
WTMTSCORE	INT	Minimum weighted match score needed to make a match decision.
NUMNAMEMUSTMT	CHAR (1)	Number of names that must match. Values are O (one name) or A (all names).
CMPFTOMNAME	CHAR (1)	Specifies the setting for the <i>Compare Given Name1 to Given Name2</i> option. (Y/N)
MTONHYPLNAME	CHAR (1)	Specifies the setting for the <i>Match on hyphenated family name</i> option. (Y/N)
TRNONMAIDENADJ	CHAR (1)	Specifies the setting for the <i>Ignore family name when female</i> option. (Y/N)
IGNFIRMIFNAME	CHAR (1)	Specifies the setting for the <i>Ignore Firm if Name matches</i> option. (Y/N)
IGNORESTIFBOX	CHAR (1)	Specifies the setting for the <i>Match on Street and RR, or on Box</i> option. (Y/N)
ADDRBLMIFFIRM	CHAR (1)	Specifies the setting for the <i>Address matches blank if Firms match</i> option. (Y/N)
UNIQRESRRNOBOX	CHAR (1)	Specifies the setting for the <i>Unique on resident if RR, but no Box</i> option. (Y/N)

Note


The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.


Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.4.6 MTCRITDEF

This table contains match criteria information for each match set and is applicable to the Match transform. The information is used in the Match Input Fields and Detailed Criteria Definition sections of the Match Criteria Summary.

Column	Data type definition	Description
 CMPCCRITID	INT	Sequential number that identifies the criteria of a match level.

Column	Data type definition	Description
 CRITID	INT	Sequential number that identifies an individual criteria.
CRITNAME	NVARCHAR (256)	Name given to the match criteria when it was created.
KEYID	INT	Sequential number that identifies the type of a key field.
MTSCORE	INT	Threshold for similarity scores. Similarity scores at or above this setting are considered a match.
NOMTSCORE	INT	Threshold for similarity scores. Similarity scores at or below this setting are not considered a match.
ONEFLDBLNKOP	CHAR (1)	Setting for the One Field Blank operation. Values are E (evaluate) and I (ignore).
ONEFLDBLNKSCORE	INT	Score given to the criteria when one of the two fields compared is blank.
BTFLDBLNKOP	CHAR (1)	Setting for the Both Fields Blank operation. Values are E (evaluate) and I (ignore).
BTFLDBLNKSCORE	INT	Score given to the criteria when both fields compared are blank.
CNTRBTOWTSCORE	INT	Weight percentage given to the match criteria's contribution to the weighted score.
USEINWTSCOREIFGT	INT	Minimum similarity score needed to qualify the match criteria for use in determining the weighted match score.
ZWTSCOREIFLTREQ	INT	Minimum similarity score needed for the match criteria to qualify for contributing a value other than zero to the weighted match score.
CMPALGO	CHAR (1)	String comparison algorithm that was used. Values are F for field or W for word.
CHKTRANSPTSLET	CHAR (1)	Indicates whether to check for transposed letters (Y/N).
INITADJSCORE	INT	Adjustment score given when fields with whole words match to fields with initials.
SUBSTRADJSCORE	INT	Adjustment score given when fields with longer strings of words match to fields with shorter strings of words (the shorter string must match the first part of the longer string).
APPRSUBADJSCORE	INT	Adjustment score given when fields with longer strings of words do not match to fields with shorter strings of words (the shorter string does not match the first part of the longer string).
ABBRADJSCORE	INT	Adjustment score given to the abbreviation substring adjustment score when the first letter of the shorter word matches the first letter of the longer word, and all remaining letters of the shorter word appear in the longer word in the same order as in the shorter word.

Column	Data type definition	Description
EXTABBRADJSORE	INT	Adjustment score given when two fields match based on a combination of the abbreviation adjustment score. Keep in mind the following requirements for the extended abbreviation adjustment: <ul style="list-style-type: none"> The first letter of the short word must match the first letter of the first word in the multiple-word string. The remaining letters of the short word must be found in order in the multiple-word string. Letters that match are given a score of 100. The remaining letters are given the score that you specify. The two scores are proportionally combined to render the overall score.
NUMWDMTEXTOPT	CHAR (1)	Numeric word setting. Values include N (none), A (any position), S (same position), P (any position consider punctuation), or Y (any position ignore punctuation).

Note



The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.4.7 MTDUPESDATA

This table contains information about duplicate records as a sample of the match results and is applicable to the Match transform. The information is used for the Match Duplicate Sample Report.

Columns	Data type definition	Description
 RECNUM	INT	Sequential number assigned to each sample record output.
 NAME	NVARCHAR (128)	Name of the field listed.
VALUE	NVARCHAR (256)	Value in the field listed.

Note



The OBJECT_KEY column is included in this table but it is not listed. It is a primary key common to most tables. The OBJECT_ID column is not in this table.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.4.8 MTGSSRCBYSRCSTS

This table contains the inter-match and intra-match source counts for the *Source by Source Statistics* section of the Match Source Stats Summary report. This table is applicable to the Match transform.

Column	Data type definition	Description
 SRCID	INT	Sequential number that identifies a source.
 PROCID	INT	Sequential number that identifies a match level or an association.
GSNAME	NVARCHAR (15)	Group statistics name.
SRCNAME	NVARCHAR (256)	Source name.
OTHERSRCNAME	NVARCHAR (256)	Other source name.
TOTDUPES	INT	Total matches between the source and the other source.

i Note



The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.4.9 MTGSSRCSTS

This table contains information about the distribution of the duplicates found in various input source records and is applicable to the Match transform. It includes information about how the duplicates were distributed as master records and subordinate records. The information is used for the Match Source Statistics Report.

Column	Data type definition	Description
 SRCID	INT	Sequential number that identifies a source.
 PROCID	INT	Sequential number that identifies a match level or an association.

Column	Data type definition	Description
GSNAME	NVARCHAR (15)	Group statistics name.
SRCNAME	NVARCHAR (256)	Source name.
SSSUB	INT	Single source subordinate record count for this source.
MSSUB	INT	Multiple source subordinate record count for this source.
SSMASTS	INT	Single source master record count for this source.
MSMASTS	INT	Multiple source master record count for this source.
NUMRECS	INT	Record count for this source.

Note

The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.4.10 MTINFO

This table contains the match set name and is applicable to the Match transform. The information is used in all of the Match reports except the Best Record Summary report.

Column	Data type definition	Description
MTSET	NVARCHAR (256)	Name of the match set.

Note







The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.4.11 MTINSRCBYSRC

This table contains information about how often each input source matched the other input sources, and these statistics are generated by the Match transform. This table is populated once per input source group statistics object if statistics are enabled. This table is similar to the MTGSSRCBYSRCSTS table. To obtain the numbers for the source groups, the rows with the same INSRCGRPID and the same INSRCGRPID_OTHER must be added.

Column	Data type definition	Description
 PROCID	INT	Sequential number used to identify a match level or an association.
 INSRCID	INT	Sequential number used to identify an input source.
 INSRCID_OTHER	INT	Sequential number used to identify an input source that had one or more records that matched one or more records in INSRCID.
 INSRCGRPID	INT	Sequential number used to identify an input source group. A value of zero implies records that do not belong to an input source group.
 INSRCGRPID_OTHER	INT	Sequential number used to identify the input source group of INSRCID_OTHER. A value of zero implies that INSRCID_OTHER does not belong to an input source group.
 OBJNAME	NVARCHAR (15)	Name of the input source group statistics object.
MATCHES	INT	Sequential number used to identify the input source group of INSRCID. A value of zero indicates that INSRCID does not belong to an input source group.

Note


The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.4.12 MTINSRCGRPINFO

This table contains input source group information and is applicable to the Match transform. It is populated once per transform when statistics are enabled and the source groups section of the input source object is defined.

Column	Data type definition	Description
 INSRGRPID	INT	Sequential number that identifies an input source group.
INSRCGRPNAME	NVARCHAR (255)	Name of an input source group.
MATCHED_COUNT	INT	Number of input records that belong to this input source group because they matched a specified input source.
DEFAULT_COUNT	INT	Number of input records that belong to this input source group because it is the default input source group.

If an input source exists that does not belong to a super source, that input source is assigned a super source ID of zero. A corresponding entry is made into the MTINSRCGRPINFO table in the following fields:

- INSPRSRCID
- INSPRSRCNAME
- MATCHED_COUNT
- DEFAULT_COUNT

All of these fields are set to zero. This allows the report code to successfully perform join operations.

i Note



The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.4.13 MTINSRCINFO

This table contains input source information and is applicable to the Match transform. It is populated once per transform if statistics are enabled and the input source object is defined.

Column	Data type definition	Description
 INSRCID	INT	Sequential number used to identify the input source.
 INSRGRPID	INT	Sequential number used to identify the input source group. The number is zero if this input source does not belong to a source group.
INSRCNAME	NVARCHAR (255)	Name of the input source.
INSRCVALUE	NVARCHAR (255)	Value of the input source.
INSRCTYPE	NVARCHAR (8)	Input source type. Valid values are NORMAL, SUPPRESS, and SPECIAL.

Column	Data type definition	Description
MATCHED_COUNT	INT	Number of input records that belong to this input source because their input source value matched the value of this source.
DEFAULT_COUNT	INT	Number of records that belong to this input source because it is the default input source.

Note





The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.4.14 MTINSRCMSRC

This table contains the multi-source statistics of each input source and each source group and is applicable to the Match transform. The table is populated once per input source group statistics object if statistics are enabled. To get the numbers for the source groups, the rows with the same INSRCGRPID must be added.

Column	Data type definition	Description
 PROCID	INT	Sequential number that identifies a match level or an association.
 INSRCID	INT	Sequential number that identifies an input source.
 INSRCGRPID	INT	Sequential number that identifies an input source group. When INSRCGRPID is zero, the record does not belong to a source group.
 OBJNAME	NVARCHAR (15)	Name of the input source group statistics object.
SRC2	INT	Number of records in INSRCID that are masters in match groups with a source count of 2.
SRC3	INT	Number of records in INSRCID that are masters in match groups with a source count of 3.
SRC4	INT	Number of records in INSRCID that are masters in match groups with a source count of 4.
SRC5	INT	Number of records in INSRCID that are masters in match groups with a source count of 5.
SRC6	INT	Number of records in INSRCID that are masters in match groups with a source count of 6.

Column	Data type definition	Description
SRC7	INT	Number of records in INSRCID that are masters in match groups with a source count of 7.
SRC8	INT	Number of records in INSRCID that are masters in match groups with a source count of 8.
SRC9	INT	Number of records in INSRCID that are masters in match groups with a source count of 9.
SRC10	INT	Number of records in INSRCID that are masters in match groups with a source count of 10.

Note






The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.4.15 MTINSRCSELECT

This table contains input source group information and is applicable to the Match transform. This table is populated once per input source select record object if statistics are enabled. To obtain the numbers for the source groups, the rows with the same INSRGRPID must be added. Refer to the *Reference Guide, Output flag selection options* section for more information.

Column	Data type definition	Description
 PROCID	INT	Sequential number that identifies the match level or an association.
 INSRCID	INT	Sequential number that identifies an input source.
 INSRGRPID	INT	Sequential number that identifies an input source group. When INSRGRPID is zero, the record does not belong to a source group.
 SECSRCID	INT	Currently not used.
 OBJNAME	NVARNVARCHAR (15)	Name of the input source group statistics object.
SECSRCNAME	NVARNVARCHAR (255)	Currently not used.
SSMASTERS_SEL	NVARNVARCHAR (1)	Indicates whether the <i>Single source masters</i> option is selected (Y/N).

Column	Data type definition	Description
SSMASTERS_KEEP	INT	Number of single source master records in INSRCID that are flagged for selection. Depending on the input source type, these are either normal or special records.
SSMASTERS_DROP	INT	Number of single source master records in INSRCID that are not flagged for selection. Depending on the input source type, these are either normal or special records.
MSMASTERS_SEL	NVARCHAR (1)	Indicates whether the <i>Multiple source masters</i> option is selected (Y/N).
MSMASTERS_KEEP	INT	Number of multi-source master records in INSRCID that are flagged for selection. Depending on the input source type, these are either normal or special records.
MSMASTERS_DROP	INT	Number of multi-source master records in INSRCID that are not flagged for selection. Depending on the input source type, these are either normal or special records.
SSSUBS_SEL	NVARCHAR (1)	Indicates whether the <i>Single source subordinates</i> option is selected (Y/N).
SSSUBS_KEEP	INT	Number of single source subordinate records in INSRCID that are flagged for selection. Depending on the input source type, these are either normal or special records.
SSSUBS_DROP	INT	Number of single source subordinate records in INSRCID that are not flagged for selection. Depending on the input source type, these are either normal or special records.
MSSUBS_SEL	NVARCHAR (1)	Indicates whether the <i>Multiple source subordinates</i> option is selected (Y/N).
MSSUBS_KEEP	INT	Number of multi-source subordinate records in INSRCID that are flagged for selection. Depending on the input source type, these are either normal or special records.
MSSUBS_DROP	INT	Number of multi-source subordinate records in INSRCID that are not flagged for selection. Depending on the input source type, these are either normal or special records.
SUPSUBS_SEL	NVARCHAR (1)	Indicates whether the <i>Suppression subordinates</i> option is selected (Y/N).
SUPSUBS_KEEP	INT	Number of suppress subordinate records in INSRCID that are flagged for selection.
SUPSUBS_DROP	INT	Number of suppress subordinate records in INSRCID that are not flagged for selection.
UNIQUES_SEL	NVARCHAR (1)	Indicates whether the <i>Unique</i> option is selected (Y/N).
UNIQUES_KEEP	INT	Number of unique records in INSRCID that are flagged for selection. Depending on the input source type, these are either normal or special records.
UNIQUES_DROP	INT	Number of unique records in INSRCID that are not flagged for selection. Depending on the input source type, these are either normal or special records.
SUPMASTERS_SEL	NVARCHAR (1)	Indicates whether the <i>Suppression masters</i> option is selected (Y/N).

Column	Data type definition	Description
SUPMASTERS_KEEP	INT	Number of suppress master records in INSRCID that are flagged for selection.
SUPMASTERS_DROP	INT	Number of suppress master records in INSRCID that are not flagged for selection.
SUPMATCHES_SEL	NVARCHAR (1)	Indicates whether the <i>Suppression matches</i> (normal and special records that follow a suppress record in a match group) option is selected (Y/N).
SUPMATCHES_KEEP	INT	Number of suppress match records in INSRCID that are flagged for selection.
SUPMATCHES_DROP	INT	Number of suppress match records in INSRCID that are not flagged for selection.
SUPUNIQUES_SEL	NVARCHAR (1)	Indicates whether the <i>Suppression uniques</i> option is selected (Y/N).
SUPUNIQUES_KEEP	INT	Number of unique suppress records in INSRCID that are flagged for selection.
SUPUNIQUES_DROP	INT	Number of unique suppress records in INSRCID that are not flagged for selection.

i Note

The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information


[Repository tables common columns \[page 1297\]](#)




12.2.4.16 MTINSRCSTATS

This table contains the statistics of each input source and each source group and is applicable to the Match transform. It is populated once per input source group statistics object if statistics are enabled.

i Note

To obtain the numbers for the source groups, the rows with the same INSRCGRPID must be added.

Column	Data type definition	Description
 PROCID	INT	Sequential number that identifies a match level or an association.

Column	Data type definition	Description
 INSRCID	INT	Sequential number that identifies an input source.
 INSRCGRPID	INT	Sequential number that identifies an input source group. When the INSRCGRPID value is zero, the record does not belong to a source group.
 OBJNAME	NVARCHAR (15)	Name of the input source group statistics object.
SSMASTERS	INT	Number of records in INSRCID that are single source masters. Depending on the input source type, these could be normal or special records.
MSMASTERS	INT	Number of records in INSRCID that are multi source masters. Depending on the input source type, these could be normal or special records.
SSSUBS	INT	Number of records in INSRCID that are single source subordinates. Depending on the input source type, these could be normal or special records.
MSSUBS	INT	Number of records in INSRCID that are multi source subordinates. Depending on the input source type, these could be normal or special records.
SUPSUBS	INT	Number of records in INSRCID that are suppress subordinates.
UNIQUES	INT	Number of records in INSRCID that are unique. Depending on the input source type, these could be normal or special records.
SUPMASTERS	INT	Number of records in INSRCID that are suppress master records.
SUPMATCHES	INT	Number of records in INSRCID that are suppress matches (normal and special records that follow a suppress record in a match group).
SUPUNIQUES	INT	Number of suppress records in INSRCID that are unique.

Note


The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.



Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.4.17 MTKEYDEF

This table contains match criteria information for each match set and is applicable to the Match transform. The information is used in the Match Criteria Summary report.

Column	Data type definition	Description
 KEYID	INT	Number that represents the field type of the key field.

Column	Data type definition	Description
 KEYFLDNUM	INT	Number that indicates which occurrence of the key field this is. The first occurrence will have a value of 1, the second will have a value of 2, and so on. An example of a key field that supports multiple occurrences is the Given Name field, where each occurrence is a different person.
 KEYFLDALT- NUM	INT	Number that indicates which alternate (match standard) of the key field this is. The original data will have a value of 0, the first alternate will have a value of 1, the second alternate will have a value of 2, and so on.
KEYNAME	NVARCHAR (256)	Name assigned to the key field when it was created in the job.
KEYLEN	INT	Number of characters in the database field that are used in creating the key field.
DATAECFLDNAME	NVARCHAR (256)	Field's input mapped name.
RMVPUNCT	CHAR (1)	Indicates whether the field was preprocessed by removing punctuation (Y/N).
CONVTOUPPER	CHAR (1)	Indicates whether the field was preprocessed by converting text to upper case (Y/N).
STDDIACHRS	CHAR (1)	Indicates whether the field was preprocessed by converting diacritical characters (Y/N).
CONVTXTTONUM	CHAR (1)	Indicates whether the field was preprocessed by converting numbers represented by text to numerals (Y/N).

Note


The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.4.18 MTPROCESS

This table contains information about the Match transform processing and is applicable to the Match transform. The information is used in all of the Match reports except the Best Record Summary report.

Column	Data type definition	Description
 PROCID	INT	Sequential number that identifies a match level or an association.
PROCTYPE	CHAR (1)	Process type. Valid values are B (Break Group Process), M (Match Level Process), or A (Association Process).
PROCNAME	NVARCHAR (15)	Process name.

i Note




The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.2.4.19 MTRULESRES

This table contains information about the effect of the criteria on the total matching process and is applicable to the Match transform. The information is used in the Match Contribution Report (Criteria Information subreport).

Columns	Data type definition	Description
 PROCID	INT	Sequential number that identifies a match level or an association.
 CMPCRITID	INT	Sequential number that identifies the criteria of a match level.
 CRITID	INT	Sequential number that identifies an individual criteria. A value of 999 is used for the weighted results.
MTDECS	INT	Number of match decisions made by this criteria.
NOMTDECS	INT	Number of no match decisions made by this criteria.
CTDECS	INT	Number of records that were allowed to continue with the match process after a no match decision was made.

i Note

The OBJECT_KEY and OBJECT_ID columns are included in this table but they are not listed. They are primary keys common to most tables.

Related Information

[Repository tables common columns \[page 1297\]](#)

12.3 Data quality statistics tables and supplemental content information

The data quality statistics tables and the supplemental content file enable you to analyze many aspects of the quality of your data.

The data quality statistics tables in the repository contain summary and record-level statistics about the cleansing and assignment processes performed on your data. You use the supplemental content information file (TASK_LOCALIZATION) to translate values that appear in the statistics tables into meaningful information.

You can enable data quality statistics in the Data Cleanse, Geocoder, and Global Address Cleanse transforms. The software sends the statistics to the applicable repository table.

There are two main reasons for generating and using the data quality statistics information:

- To view aggregated results and summary information about the quality of your data.
- To drill into records to gain information about how the software cleansed specific data elements.

Data quality statistics tables

Table name	Transform	Description
CLEANSE_ADDRESS_RECORD_INFO_	Global Address Cleanse	Analyze this non-summary table to gain information about the results of the address cleanse process on the data. The table contains statistics that focus on the results of address cleansing, which includes assignment information and assignment type for each address. This table is populated only when the transform is set for global address cleansing, and only for the address data.
CLEANSE_CHANGE_INFO_	Data Cleanse Global Address	Analyze this non-summary table to identify areas of concern in your source data by the number of significant changes. For example, you may decide to research a data source further if the data shows a high count of significant changes for firm or address components.
CLEANSE_COMPONENT_INFO_ (available after 4.2.6.0 release)	Data Cleanse Global Address Cleanse	This non-summary table contains position information for each specified data element when it is parsed from an input field and/or written to an output field.

Table name	Transform	Description
CLEANSE_INFO_CODES_	Data Cleanse Global Address Cleanse	Analyze this non-summary table to find potential suspect data in your data source. You can analyze the table information to view a distribution of information codes or a count of missing, suspect, or blank data.
CLEANSE_STATISTICS_	Data Cleanse Global Address Cleanse	This summary table contains data for each unique ENTITY_ID for use in providing a high-level insight into what the software changed during the cleansing process.
GEOCODE_INFO_CODES_	Geocoder	Analyze this non-summary table to track the reasons why records were not assigned the highest level of latitude and longitude by using the information code data.
GEOCODE_STATISTICS_	Geocoder	Analyze this summary table to view a summary of the geocode process on the data. The table provides the total number of rows in the data and the total number of rows that were assigned a latitude and longitude.

Related Information

[CLEANSE_ADDRESS_RECORD_INFO_ \[page 1352\]](#)

[CLEANSE_CHANGE_INFO_ \[page 1356\]](#)

[CLEANSE_COMPONENT_INFO_ \[page 1360\]](#)

[CLEANSE_INFO_CODES_ \[page 1366\]](#)

[CLEANSE_STATISTICS_ \[page 1370\]](#)

[GEOCODE_INFO_CODE_ \[page 1373\]](#)

[GEOCODE_STATISTICS_ \[page 1376\]](#)

12.3.1 Content information for data quality statistics tables

You must use the supplemental TASK_LOCALIZATION file to translate values that appear in the statistics tables into meaningful information.

You need the task localization file to translate the data into recognizable words and descriptions. Download the localization file named `dqs_task_localization_<language>.csv` from the SAP Community Network. For instructions see [Downloading the data quality statistics task localization file \[page 444\]](#).

You can customize the translation file to reflect words and descriptions that are more recognizable to your business. After you download the task localization file, you are responsible for storing and preserving the file. If you customize it, you need to transfer those customizations to any future updates.

The following table contains descriptions of the contents of the TASK_LOCALIZATION file. After this table, find the task localization key file for explanations of the characters and codes that may appear in the data quality statistics tables.

TASK_LOCALIZATION file contents

Column name	Data type definition	Description
LOC_TYPE_ID	INT	Type being localized. Use this column to join with the data quality statistics for the description of the LOC_ID.
LOC_ID	NVARCHAR(255)	ID you want to localize. Use this column to join with the data quality statistics for the description of the ID.

Column name	Data type definition	Description																										
LANGUAGE	NVARCHAR(1)	<p>Identifies which language the information should be translated into. Each of the downloaded files include the language of the translation file. If multiple languages are supported the language key (below) can be used to return the correct translation. For example if you require English, use E, if German use D, and so on.</p> <p>Language key</p> <table><tr><th>Key</th><th>Language</th></tr><tr><td>1</td><td>Chinese</td></tr><tr><td>3</td><td>Korean</td></tr><tr><td>D</td><td>German</td></tr><tr><td>E</td><td>English</td></tr><tr><td>F</td><td>French</td></tr><tr><td>J</td><td>Japanese</td></tr><tr><td>K</td><td>Danish</td></tr><tr><td>N</td><td>Dutch</td></tr><tr><td>P</td><td>Portuguese</td></tr><tr><td>R</td><td>Russian</td></tr><tr><td>S</td><td>Spanish</td></tr><tr><td>T</td><td>Turkish</td></tr></table>	Key	Language	1	Chinese	3	Korean	D	German	E	English	F	French	J	Japanese	K	Danish	N	Dutch	P	Portuguese	R	Russian	S	Spanish	T	Turkish
Key	Language																											
1	Chinese																											
3	Korean																											
D	German																											
E	English																											
F	French																											
J	Japanese																											
K	Danish																											
N	Dutch																											
P	Portuguese																											
R	Russian																											
S	Spanish																											
T	Turkish																											
DESCRIPTION	NVARCHAR(255)	Localized value																										

The following table provides explanations for the characters and codes that may appear in the data quality statistics tables. Use the information as an aid to understand the contents of the various data quality statistics tables.

TASK_LOCALIZATION key

LOC_TYPE_ID	Attribute	Statistics table	Transform	Notes
1	ENTITY_ID	CLEANSE_INFO_CO- DES_	Data Cleanse LOC_ID = 2-7	Identifies which entity type the information pertains to.
		CLEANSE_CHANGE_I NFO_	Global Address Cleanse LOC_ID = 1	Not all transform input fields apply to data quality statistics. See Data quality statistics transform set up re- quirements [page 451] for more information.
		CLEANSE_STATIS- TICS_		
		CLEANSE_COMPO- NENT_INFO_		
2	INFO_CODE	CLEANSE_INFO_CO- DES_	Data Cleanse Global Address Cleanse	Data Cleanse and Global address cleanse information codes. The code that identi- fies the rows that may require manual review because the data is suspect.

i Note

Not all existing in-
formation codes
apply to data qual-
ity statistics.

See [Information
codes \(Global Ad-
dress Cleanse\)
\[page 808\]](#) for de-
scriptions of
Global Address
Cleanse informa-
tion codes.

See [Detailed de-
scriptions of infor-
mation codes
\[page 829\]](#) for de-
scriptions of Data
Cleanse informa-
tion codes.

LOC_TYPE_ID	Attribute	Statistics table	Transform	Notes
3	ISO_COUNTRY_2CHAR	CLEANSE_ADDRESS_INFORMATION	Global Address Cleanse	<p>Global Address Cleanse ISO two-character country code.</p> <p>See Country ISO codes and assignment engines [page 791] for information about ISO country codes.</p>
4	ASSIGNMENT_TYPE	CLEANSE_ADDRESS_INFORMATION	Global Address Cleanse	<p>Global Address Cleanse output address type.</p> <p>A one- or two-character code that represents the type of address.</p> <p>See Output fields for the Global Address Cleanse transform [page 608] for information about the Address_Type output field.</p>
5	ASSIGNMENT_INFORMATION	CLEANSE_ADDRESS_INFORMATION	Global Address Cleanse	<p>Global Address Cleanse output assignment information.</p> <p>Indicates whether a record is valid, invalid, or corrected based on the status and information codes.</p> <p>See Output fields for the Global Address Cleanse transform [page 608] for information about the Assignment_Type output field.</p>

LOC_TYPE_ID	Attribute	Statistics table	Transform	Notes
6	ASSIGNMENT_LEVEL	CLEANSE_AD- DRESS_RE- CORD_INFO_	Global Address Cleanse	<p>Global Address Cleanse assignment level.</p> <p>The level to which this transform matched the address to the data in the reference files (di- rectories).</p> <p>See Output fields for the Global Address Cleanse transform [page 608] for infor- mation about the As- signment_Level output field.</p>
7	COMPONENT_ID	CLEANSE_CHANGE_I NFO_ CLEANSE_COMPO- NENT_INFO_	Data Cleanse Global Address Cleanse	<p>Identification number for a specific portion of the data entity. For ex- ample, an address en- tity has components of street, street number, city, postcode, and so on.</p>
8	COMPONENT_ELE- MENT_ID	CLEANSE_CHANGE_I NFO_ CLEANSE_COMPO- NENT_INFO_	Data Cleanse Global Address Cleanse	<p>Identification number for an element within a COMPONENT_ID. For example, a city compo- nent identified by a LOC_TYPE_ID = 7 and LOC_ID of 105 has the following possible ele- ments:</p> <ul style="list-style-type: none"> • City name = 1 • City description = 2 • City qualifier = 3 • City code = 4 • City addition = 5

LOC_TYPE_ID	Attribute	Statistics table	Transform	Notes
9	CHANGE_CATEGORY_ID	CLEANSE_CHANGE_INFO_	Data Cleanse Global Address Cleanse	<p>Identification number for the type of update performed on the output field:</p> <ul style="list-style-type: none"> • Change = 12 • Add = 13 • Delete = 14
10	CHANGE_SIGNIFICANCE_ID	CLEANSE_CHANGE_INFO_	Data Cleanse Global Address Cleanse	Identification number for the change significance level.
11	INFO_CODE	GEOCODE_INFO_CODES_	Geocoder	<p>Geocoder information codes.</p> <p>A three-character code that provides information about the geocoding results.</p> <div> <p>i Note</p> <p>Not all existing information codes apply to data quality statistics.</p> <p>See Information codes [page 548] for descriptions of Geocode information codes.</p> </div>
12	REVIEW_GROUP	MATCH_GROUP_INFO_	Match	Not currently available
13	CONFLICT_GROUP	MATCH_GROUP_INFO_	Match	Not currently available
14	DST_ROW_TYPE	BEST_RECORD_RESULTS_	Best Record	Not currently available
15	OPERATION_TYPE	BEST_RECORD_RESULTS_	Best Record	Not currently available

Related Information

[Data quality statistics tables and supplemental content information \[page 1344\]](#)

[Downloading the data quality statistics task localization file \[page 444\]](#)

12.3.2 CLEANSE_ADDRESS_RECORD_INFO_

Provides information about address fields in your data that are changed because of the address cleansing process.

- Contains up to two rows of statistics for each address entity
- Available for Global Address Cleanse transform
- Contains record-level statistics (non-summary)

Analyze this non-summary table to retrieve information about the results of the address cleanse process on the data. The table contains statistics that focus on the results of address cleansing, which includes assignment details for the address.

Download the task localization file, `dqs_task_localization_<language>.csv` from the SAP Community Network website at <http://scn.sap.com/docs/DOC-68523> to use for translating values in tables.

Column name	Data type definition	Description
OBJECT_KEY	INT	This unique internal ID is assigned at the time of execution for each of the objects created in Data Services. This is a foreign key to the JOB_KEY in the DQVW_AGR_RPTSTATS and DQVW_RE-PORTS_STAT tables.
OBJECT_ID	NVARCHAR(255)	This ID is unique for each operation within a Data Services work flow. This is a foreign key to the OBJECT_ID in the DQVW_AGR_RPTSTATS and DQVW_RE-PORTS_STAT tables.
TABLE_ID	INT	Set to 1.

Column name	Data type definition	Description
ROW_ID	INT	An internal identification generated by the transform that uniquely identifies a row processed by that transform. Appears in each non-summary data quality statistics table and is output by the transform. Use this field to join the non-summary data quality statistics tables to the transform output. Subsequently this can be joined to the input source using the input source primary key if it is included in the transform output.
ENTITY_INSTANCE	INT	A number to describe which entity from the record the entry pertains to. For example the ENTITY_INSTANCE for address can be 1-2, and for person 1-6. The ENTITY_INSTANCE corresponds to the instance of the input mapping of an entity, and is therefore dependent on the order of input column mapping.
ENTITY_INSTANCE_OCCURRENCE	INT	A number to distinguish entities when multiple entities are encountered in the same entity instance. For example this number will distinguish between two person names when a dual name is input in a person column or between two addresses when a dual address is input in a single group of address columns.
DATA_SOURCE	NVARCHAR(255)	The name of the data source that the record comes from. The value is input from the column input mapped to DATA_SOURCE_ID and is simply passed through to the data quality statistics table.
ISO_COUNTRY_2CHAR	VARCHAR(4)	This is the Global Address Cleanse ISO two-character country code. This is a foreign key to the LOC_ID column in the TASK_LOCALIZATION table where the LOC_TYPE_ID = 3.

Column name	Data type definition	Description
ASSIGNMENT_TYPE	VARCHAR(4)	A one or two character code that represents the type of address. This is a foreign key to the LOC_ID column in the TASK_LOCALIZATION table where the LOC_TYPE_ID = 4.
ASSIGNMENT_INFORMATION	VARCHAR(4)	Indicates whether a record is valid, invalid, corrected, or blank based on the status and information codes. This is a foreign key to the LOC_ID column in the TASK_LOCALIZATION table where the LOC_TYPE_ID = 5.
ASSIGNMENT_LEVEL	VARCHAR(4)	The level to which this transform matched the address to the data in the reference files. This is a foreign key to the LOC_ID column in the TASK_LOCALIZATION table where the LOC_TYPE_ID = 6.

❖ Example

Use the DQVW_AGR_RPTSTATS and DQVW_REPORTS_STAT tables to identify the unique key for the results in the data quality statistics tables for the query. The unique key of OBJECT_KEY and OBJECT_ID will be used to analyze the data in the data quality statistic tables. In this example the unique key is:

C.OBJECT_KEY = 412

AND

C.OBJECT_ID = "4f2ba82f-3e7c-44ce-8543-8f5844b44f60"

≡ Sample Code

```
SELECT
    L.DESCRPTION AS "Country",
    COUNT(C.ROW_ID) AS "TOTAL"
FROM
    CLEANSE_ADDRESS_RECORD_INFO_ AS C,
    TASK_LOCALIZATION AS L
WHERE
    -- Get the statistics for the desired job and transform.
    C.OBJECT_KEY = 412 AND
    C.OBJECT_ID = "4f2ba82f-3e7c-44ce-8543-8f5844b44f60" AND
    -- Get the ISO_COUNTRY_2CHAR description.
    L.LOC_TYPE_ID = 3 AND
    C.ISO_COUNTRY_2CHAR = L.LOC_ID
GROUP BY
    L.DESCRPTION
ORDER BY
    TOTAL DESC
```

Results:

Country	Total
Bahamas	642
Italy	323
United States	36
Austria	19
Canada	19
France	16
United Kingdom	15
Germany	11
China	6
Brazil	6
Poland	6
Belgium	5
Puerto Rico	4
Australia	4
Czechia	3
India	2
Netherlands	2
Portugal	2
Republic of Korea	1
Isle of Man	1
Sweden	1
Mexico	1
Japan	1
Switzerland	1
New Zealand	1
Finland	1

12.3.3 CLEANSE_CHANGE_INFO_

Provides a non-summary table that includes a row for each significant change identified by the cleanse process.

- Contains a row of statistics for each entity that was significantly changed during the cleanse process
- Applicable for the Data Cleanse and Global Address Cleanse transforms
- Contains record-level statistics (non-summary)




A significant change is a change that is made during the cleansing process that a user may want to review to ensure the data was cleansed as expected.





Currently the table identifies only high significance changes. An example of a high significant change is when a postcode1 for an address is deleted, or a change in the last name such as “Sanches” was input and output as “Sanchez”.





See the data quality statistics table, CLEANSE_STATISTICS_ ([CLEANSE_STATISTICS_ \[page 1370\]](#)), to analyze the total counts for all rows of the input data that were identified as significantly changed.

Analyze this non-summary table to identify concerning areas of your source data by the number of significant changes. For example, you may decide to research a data source further if the data shows a high count of significant changes for firm or address components.

Download the task localization file, `dqs_task_localization_<language>.csv` from the SAP Community Network website at <http://scn.sap.com/docs/DOC-68523> to use for translating values in tables.

Column name	Data type definition	Description
 OBJECT_KEY	INT	This unique internal ID is assigned at the time of execution for each of the objects created in Data Services. This is a foreign key to the JOB_KEY in the DQVW_AGR_RPTSTATS and DQVW_RE-PORTS_STAT tables.
 OBJECT_ID	NVARCHAR(255)	This ID is unique for each operation within a Data Services work flow. This is a foreign key to the OBJECT_ID in the DQVW_AGR_RPTSTATS and DQVW_RE-PORTS_STAT tables.
 TABLE_ID	INT	Set to 1.

Column name	Data type definition	Description
 ROW_ID	INT	An internal identification generated by the transform that uniquely identifies a row processed by that transform. Appears in each non-summary data quality statistics table and is output by the transform. Use this field to join the non-summary data quality statistics tables to the transform output. Subsequently this can be joined to the input source using the input source primary key if it is included in the transform output.
 ENTITY_ID	NVARCHAR(12)	Identification number that refers to entities such as person, firm, and address. This is a foreign key to the LOC_ID column in the TASK_LOCALIZATION table where the LOC_TYPE_ID = 1.
 ENTITY_INSTANCE	INT	A number to describe which entity from the record the entry pertains to. For example the ENTITY_INSTANCE for address can be 1-2, and for person 1-6. The ENTITY_INSTANCE corresponds to the instance of the input mapping of an entity, and is therefore dependent on the order of input column mapping.
 ENTITY_INSTANCE_OCCURRENCE	INT	A number to distinguish entities when multiple entities are encountered in the same entity instance. For example this number will distinguish between two person names when a dual name is input in a person column or between two addresses when a dual address is input in a single group of address columns.
DATA_SOURCE	NVARCHAR(255)	The name of the data source that the record comes from. The value is input from the column input mapped to DATA_SOURCE_ID and is simply passed through to the data quality statistics table.

Column name	Data type definition	Description
 COMPONENT_ID	NVARCHAR(12)	Identification number that refers to data components. For example, the entity address has components of street, street number, city, and postcode. This is a foreign key to the LOC_ID column in the TASK_LOCALIZATION table where the LOC_TYPE_ID = 7.
 COMPONENT_ELEMENT_ID	NVARCHAR(12)	Identification number that refers to more granular elements within a component. For example, the street component has component elements of street name, street type, street prefix, and street postfix. This is a foreign key to the LOC_ID column in the TASK_LOCALIZATION table where the LOC_TYPE_ID = 8.
 CHANGE_CATEGORY_ID	NVARCHAR(12)	Identification number that refers to the type of change that was made to the data. This is a foreign key to the LOC_ID column in the TASK_LOCALIZATION table where the LOC_TYPE_ID = 9.
 CHANGE_SIGNIFICANCE_ID	NVARCHAR(12)	Identification number that refers to the significance of the change. This is a foreign key to the LOC_ID column in the TASK_LOCALIZATION table where the LOC_TYPE_ID = 10.

❖ Example

Use the DQVW_AGR_RPTSTATS and DQVW_REPORTS_STAT tables to identify the unique key for the results in the data quality statistics tables for the query. The unique key of OBJECT_KEY and OBJECT_ID will be used to analyze the data in the data quality statistic tables. In this example the unique key is:

C.OBJECT_KEY = 412

AND

C.OBJECT_ID = "4f2ba82f-3e7c-44ce-8543-8f5844b44f60"

≡ Sample Code

```
SELECT
    C.COMPONENT_ID AS "Component ID",
    L.DESRIPTION AS "Component Description",
    COUNT (C.ROW_ID) AS "Count"
FROM
    CLEANSE_CHANGE_INFO_ AS C,
    TASK_LOCALIZATION AS L
WHERE
```



```

-- Get the statistics for the desired job and transform.
  C.OBJECT_KEY = 412    AND
  C.OBJECT_ID = "4f2ba82f-3e7c-44ce-8543-8f5844b44f60" AND
-- Get the COMPONENT_ID description.
  L.LOC_TYPE_ID = 7 AND
  C.COMPONENT_ID = L.LOC_ID
GROUP BY
  C.COMPONENT_ID, L.DESCRPTION
ORDER BY
  C.COMPONENT_ID;

```

Results:

Component ID	Component Description	Count
101	Country	971
102	Postcode	27
103	Region	7
105	City	35
106	Subcity	3
107	Subcity2	1
109	Delivery Installation	8
110	Street	43
111	Street2	2
114	House Number	2
119	Unit	1
120	Building	2
201	Person	294
202	Job Title	5
301	Organization	114
501	Phone	57
601	E-Mail	291

12.3.4 CLEANSE_COMPONENT_INFO_

Provides non-summary table that includes information for each specified data element when it is parsed from an input field and/or written to an output field.

i Note

The CLEANSE_COMPONENT_INFO_ TABLE is not available for use until after 4.2.6.0 release.

The purpose of this table is to track the exact location of individual data elements that were parsed during the cleansing process.

- Contains a row of statistics for each data element from the data source that was parsed during the cleanse process
- Applicable for the Data Cleanse and Global Address Cleanse transforms
- Contains record-level statistics (non-summary)

Position data includes the input field column start position and the data length for each applicable input component, the output field column start position and the data length for each applicable output component. Note an output column field can be populated when there is no direct link to an input, and an input may not have any directly linked output.

Analyze this non-summary table to identify the location of particular data elements found in your data during the cleansing process, enabling a review user interface to highlight strings when displaying the data before and/or after.

Download the task localization file, `dqs_task_localization_<language>.csv` from the SAP Community Network website at <http://scn.sap.com/docs/DOC-68523> to use for translating values in tables.

Column	Data type definition	Description
OBJECT_KEY	INT	This unique internal ID is assigned at the time of execution for each of the objects created in Data Services. This is a foreign key to the JOB_KEY in the DQVW_AGR_RPTSTATS and DQVW_RE-PORTS_STAT tables.
OBJECT_ID	NVARCHAR(255)	This ID is unique for each operation within a Data Services work flow. This is a foreign key to the OBJECT_ID in the DQVW_AGR_RPTSTATS and DQVW_RE-PORTS_STAT tables.
TABLE_ID	INT	Set to 1. Join TABLE_ID and COLUMN_ID to the TASK_COLUMN_DEFINITIONS_ table to obtain input field information.

Column	Data type definition	Description
ROW_ID	INT	An internal identification generated by the transform that uniquely identifies a row processed by that transform. Appears in each non-summary data quality statistics table and output by the transform. Use this field to join the non-summary data quality statistics tables to the transform output. Subsequently this can be joined to the input source using the input source primary key if it is included in the transform output.
ENTITY_ID	NVARCHAR(12)	Identification number that refers to entities such as person, firm, and address. This is a foreign key to the LOC_ID column in the TASK_LOCALIZATION table where the LOC_TYPE_ID = 1.
ENTITY_INSTANCE	INT	<p>A number to describe which entity from the record the entry pertains to. For example the ENTITY_INSTANCE for address can be 1-2, and for person 1-6.</p> <p>The ENTITY_INSTANCE corresponds to the instance of the input mapping of an entity, and is therefore dependent on the order of input column mapping.</p>
ENTITY_INSTANCE_OCCURRENCE	INT	A number to distinguish entities when multiple entities are encountered in the same entity instance. For example this number will distinguish between two person names when a dual name is input in a person column or between two addresses when a dual address is input in a single group of address columns.
DATA_SOURCE	NVARCHAR(255)	The name of the data source that the record comes from. The value is input from the column input mapped to DATA_SOURCE_ID and is simply passed through to the data quality statistics table.

Column	Data type definition	Description
COMPONENT_ID	NVARCHAR(12)	Identification number that refers to data components. For example, the entity address has components of street, street number, city, and postcode. This is a foreign key to the LOC_ID column in the TASK_LOCALIZATION table where the LOC_TYPE_ID = 7.
COMPONENT_ELEMENT_ID	NVARCHAR(12)	Identification number that refers to more granular elements within a component. For example, the street component has component elements of street name, street type, street prefix, and street postfix. This is a foreign key to the LOC_ID column in the TASK_LOCALIZATION table where the LOC_TYPE_ID = 8.
COLUMN_ID	INT	The ID of the input table's column that the component element was parsed from. When the component element spans multiple columns in the input table then there are multiple rows for the component element in this table. Join COLUMN_ID and TABLE_ID to the TASK_COLUMN_DEFINITIONS_ table to obtain input field information.
COLUMN_START_POSITION	INT	The character position in the input field where the parsed component element begins.
COLUMN_DATA_LENGTH	INT	The number of characters that was parsed from the input for the component element.
OUTPUT_TABLE_ID	INT	Set to 2. Join OUTPUT_TABLE_ID and OUTPUT_COLUMN_ID to the TASK_COLUMN_DEFINITIONS_ table to obtain output field information.

Column	Data type definition	Description
OUTPUT_COLUMN_ID	INT	The ID of the column that the component element is output to. When the component element exists in multiple output columns then there are multiple rows for the component element in this table. Join OUTPUT_COLUMN_ID and OUTPUT_TABLE_ID to the TASK_COLUMN_DEFINITIONS_ table to obtain output field information.
OUTPUT_COLUMN_START_POSITION	int	The character position in the output field where the output component element begins.
OUTPUT_COLUMN_DATA_LENGTH	INT	The number of characters in the output component element.

❖ Example

Example 1

Use the DQVW_AGR_RPTSTATS and DQVW_REPORTS_STAT tables to identify the unique key for the results in the data quality statistics tables for the query. The unique key of OBJECT_KEY and OBJECT_ID will be used to analyze the data in the data quality statistic tables. In this example the unique key is:

C.OBJECT_KEY = 412

AND

C.OBJECT_ID = "4f2ba82f-3e7c-44ce-8543-8f5844b44f60"

≡ Sample Code

```
SELECT
    C.COLUMN_ID AS "Input Column ID",
    D.COLUMN_NAME AS "Input Column Name"
FROM
    CLEANSE_COMPONENT_INFO_ AS C,
    TASK_COLUMN_DEFINITIONS_ AS D
WHERE
    -- Get the statistics for the desired job and transform.
    C.OBJECT_KEY = 412 AND
    C.OBJECT_ID = "4f2ba82f-3e7c-44ce-8543-8f5844b44f60" AND
    -- Get only the rows where something was parsed but not output.
    C.COLUMN_DATA_LENGTH > 0 AND
    C.OUTPUT_COLUMN_DATA_LENGTH = 0 AND
    -- Get the input column name.
    D.OBJECT_KEY = 412 AND
    D.OBJECT_ID = "4f2ba82f-3e7c-44ce-8543-8f5844b44f60" AND
    C.TABLE_ID = D.TABLE_ID AND
    C.COLUMN_ID = D.COLUMN_ID
GROUP BY
    C.COLUMN_ID,
    D.COLUMN_NAME
ORDER BY
    C.COLUMN_ID
```

Results:

Input Column ID	Input Column Name
2	hybrid_thread_test_3000.EMAIL1
3	hybrid_thread_test_3000.EMAILX
4	hybrid_thread_test_3000.PERSON1_FAMILY_NAME1
5	hybrid_thread_test_3000.PERSON1_GIVEN_NAME1
6	hybrid_thread_test_3000.PHONE1
7	hybrid_thread_test_3000.TITLE_LINE1
8	hybrid_thread_test_3000.PHONEX
9	hybrid_thread_test_3000.FIRM_NAME1
10	hybrid_thread_test_3000.FIRM_LINE2

❖ Example

Example 2

Use the DQVW_AGR_RPTSTATS and DQVW_REPORTS_STAT tables to identify the unique key for the results in the data quality statistics tables for the query.

The unique key of OBJECT_KEY and OBJECT_ID will be used to analyze the data in the data quality statistic tables. In this example the unique key is:

C.OBJECT_KEY = 412

AND

C.OBJECT_ID = "4f2ba82f-3e7c-44ce-8543-8f5844b44f60"

≡ Sample Code

```
SELECT
  C.ROW_ID,
  D.COLUMN_NAME AS "Input Column Name",
  L.DESCRPTION AS "Component",
  L2.DESCRPTION AS "Component Element ID",
  ABS(C.OUTPUT_COLUMN_DATA_LENGTH - C.COLUMN_DATA_LENGTH) AS "Change in
Length"
FROM
  CLEANSE_COMPONENT_INFO AS C,
  TASK_LOCALIZATION AS L,
  TASK_LOCALIZATION AS L2,
  TASK_COLUMN_DEFINITIONS AS D
WHERE
  -- Get the statistics for the desired job and transform.
  C.OBJECT_KEY = 412 AND
  C.OBJECT_ID = "4f2ba82f-3e7c-44ce-8543-8f5844b44f60" AND
```

```
-- Look for changes to length
C.COLUMN_DATA_LENGTH != C.OUTPUT_COLUMN_DATA_LENGTH AND

-- Get the name of the COMPONENT_ID
L.LOC_TYPE_ID = 7 AND
C.COMPONENT_ID = L.LOC_ID AND

-- Get the name of the COMPONENT_ELEMENT_ID
L2.LOC_TYPE_ID = 8 AND
C.COMPONENT_ID||'/'||C.COMPONENT_ELEMENT_ID = L2.LOC_ID AND

-- Get the input column name.
D.OBJECT_KEY = 412 AND
D.OBJECT_ID = "4f2ba82f-3e7c-44ce-8543-8f5844b44f60" AND
C.TABLE_ID = D.TABLE_ID AND
C.COLUMN_ID = D.COLUMN_ID
```

Results:


ROW_ID	Input Column Name	Component	Component Element	ID Change in Length
2078	DS_DQMD_AllEngines.Firm	Organization	Organization Name	14
2039	DS_DQMD_AllEngines.Firm	Organization	Organization Name	1
2026	DS_DQMD_AllEngines.Firm	Organization	Organization Name	2




12.3.4.1 TASK_COLUMN_DEFINITIONS_

Provides means to retrieve column identification information for the CLEANSE_COMPONENT_INFO_ table.

When you generate a CLEANSE_COMPONENT_INFO_ table, the software automatically creates the TASK_COLUMN_DEFINITIONS_ table as a supporting table.

The purpose of this table is to use with the CLEANSE_COMPONENT_INFO_ table to uniquely identify a table input or output, processed by a transform.

Column	Data type definition	Description
 OBJECT_KEY	INT	This unique internal ID is assigned at the time of execution for each of the objects created in Data Services. This is a foreign key to the JOB_KEY in the DQVW_AGR_RPTSTATS and DQVW_RE-PORTS_STAT tables.

Column	Data type definition	Description
 OBJECT_ID	NVARCHAR(255)	This ID is unique for each operation within a Data Services work flow. This is a foreign key to the OBJECT_ID in the DQVW_AGR_RPTSTATS and DQVW_REPORTS_STAT tables.
 TABLE_ID	INT	Input TABLE_ID =1, output TABLE_ID = 2.
 COLUMN_ID	INT	The ID of the input and output table's columns that were used.
COLUMN_NAME	NVARCHAR(128)	The input and output field names.
MAPPED_NAME	NVARCHAR(128)	The mapped name selected when the job was created for the input and output COLUMN_NAME.

See CLEANSE_COMPONENT_INFO_ data quality statistics table for examples of use ([CLEANSE_COMPONENT_INFO_ \[page 1360\]](#))

12.3.5 CLEANSE_INFO_CODES_

Provides non-summary table that includes a row for each significant information code generated by the cleanse process.

- Contains a row of statistics for each significant information code that is assigned during the cleanse process.
- Applicable for the Data Cleanse and Global Address Cleanse transforms
- Contains record-level statistics (non-summary)








Analyze this non-summary table to find potential suspect data in your data source. You can use the statistical information to view a distribution of information codes or a count of missing, suspect, or blank data.


Download the task localization file, `dqs_task_localization_<language>.csv` from the SAP Community Network website at <http://scn.sap.com/docs/DOC-68523> to use for translating values in tables.

For descriptions of information codes see [Information codes \(Global Address Cleanse\) \[page 808\]](#) and [Information codes \(Data Cleanse\) \[page 827\]](#)

Note

Not all existing information codes apply to data quality statistics.

Column	Data type definition	Description
 OBJECT_KEY	INT	This unique internal ID is assigned at the time of execution for each of the objects created in Data Services. This is a foreign key to the JOB_KEY in the DQVW_AGR_RPTSTATS and DQVW_RE-PORTS_STAT tables.
 OBJECT_ID	NVARCHAR (255)	This ID is unique for each operation within a Data Services work flow. This is a foreign key to the OBJECT_ID in the DQVW_AGR_RPTSTATS and DQVW_RE-PORTS_STAT tables.
 TABLE_ID	INT	Set to 1.
 ROW_ID	INT	An internal identification generated by the transform that uniquely identifies a row processed by that transform. Appears in each non-summary data quality statistics table and output by the transform. Use this field to join the non-summary data quality statistics tables to the transform output. Subsequently this can be joined to the input source using the input source primary key if it is included in the transform output.
 ENTITY_ID	NVARCHAR (12)	Identification number that refers to entities such as person, firm, and address. This is a foreign key to the LOC_ID column in the TASK_LOCALIZATION table where the LOC_TYPE_ID = 1.
 ENTITY_INSTANCE	INT	A number to describe which entity from the record the entry pertains to. For example the ENTITY_INSTANCE for address can be 1-2, and for person 1-6. The ENTITY_INSTANCE corresponds to the instance of the input mapping of an entity, and is therefore dependent on the order of input column mapping.
 ENTITY_INSTANCE_OCCURRENCE	INT	A number to distinguish entities when multiple entities are encountered in the same entity instance. For example this number will distinguish between two person names when a dual name is input in a person column or between two addresses when a dual address is input in a single group of address columns.

Column	Data type definition	Description
DATA_SOURCE	NVARCHAR (255)	The name of the data source that the record comes from. The value is input from the column input mapped to DATA_SOURCE_ID and is simply passed through to the data quality statistics table.
 INFO_CODE	NVARCHAR(10)	Indicates whether something potentially suspect about the data was encountered during the processing of the cleanse operation. Only records that get an information code are written to this table. Records that get multiple information codes will have multiple rows written to this table, one per information code. This is a foreign key to the LOC_ID column in the TASK_LOCALIZATION table where the LOC_TYPE_ID = 2.

❖ Example

Use the DQVW_AGR_RPTSTATS and DQVW_REPORTS_STAT tables to identify the unique key for the results in the data quality statistics tables for the query. The unique key of OBJECT_KEY and OBJECT_ID will be used to analyze the data in the data quality statistic tables. In this example this unique key is:

C.OBJECT_KEY = 412

AND

C.OBJECT_ID = "4f2ba82f-3e7c-44ce-8543-8f5844b44f60"

≡ Sample Code

```
SELECT
    C.INFO_CODE AS "Info Code",
    L.DESCRPTION AS "Info Code Description",
    COUNT (C.ROW_ID) AS "Count"
FROM
    CLEANSE_INFO_CODES AS C,
    TASK_LOCALIZATION AS L
WHERE
    -- Get the statistics for the desired job and transform.
    C.OBJECT_KEY = 412 AND
    C.OBJECT_ID = "4f2ba82f-3e7c-44ce-8543-8f5844b44f60" AND
    -- Get the description for the INFO_CODE
    L.LOC_TYPE_ID = 2 AND
    C.INFO_CODE = L.LOC_ID
GROUP BY
    C.INFO_CODE, L.DESCRPTION
ORDER BY
    C.INFO_CODE;
```

Results:

Info Code	Info Code Description	Count
I161	No entry made for job title	2335
I171	Non-title data found together with the job title	339
I221	No entry made for person name	2365
I361	No entry made for organization name	2943
I362	No entry made for organization name	2943
I711	This is not an e-mail address	344
I722	No entry made for e-mail address	2943
I731	Non e-mail data found together with the e-mail address	36
I732	Non e-mail data found together with the e-mail address	57
I751	This is not a phone number	72
I761	No entry made for phone number	13
1030	No country identified	1060
2000	Unable to identify city, region, and/or postcode information	2
2020	Unable to identify postcode. Invalid city is preventing address cleansing	965
3000	City, region, and postcode are valid, but unable to identify the street address	2
3010	City, region, and postcode are valid, but unable to match street name to directory	10
3030	House number has not been entered or is not in the directory	12
3090	An invalid or missing city is preventing address cleansing	1
3200	The building has not been entered or is not in the directory	2

Info Code	Info Code Description	Count
4000	The secondary address information has not been entered or is not in the directory	8
4500	The organization has not been entered or is not in the directory	5
5000	The address is valid, but the postal authority classifies this address as undeliverable	2

12.3.6 CLEANSE_STATISTICS_

Provides a summary of statistics for the cleansing process.

The table contains summarized data for each unique entity ID for use in providing a high-level insight into what the software changed during the cleansing process. You can use this summary table to ensure that your data was cleansed as you expected.

Information in the table

Suspect Records:

For Global Address Cleanse, this is a count of records whose information code is less than 4000, plus 5000, 5010, 5040, and 5050. Users can locate these records by looking at the information code.

A user can locate these records by using the data quality statistics table `CLEANSE_INFO_CODES_` ([CLEANSE_INFO_CODES_ \[page 1366\]](#)) to query for the information codes.

See [Information codes \(Global Address Cleanse\) \[page 808\]](#) for descriptions of Global Address Cleanse information codes.

See [Information codes \(Data Cleanse\) \[page 827\]](#) for descriptions of Data Cleanse information codes.

Blanks Records

For Global Address Cleanse, this is a count of records whose assignment information is set to "B". Users can locate these records by using the data quality statistics table `CLEANSE_ADDRESS_RECORD_INFO_` to query for the rows where `ASSIGNMENT_INFORMATION LIKE "B"`

A user can locate these records by using the data quality statistics table `CLEANSE_INFO_CODES_` ([CLEANSE_INFO_CODES_ \[page 1366\]](#)) to query for the above information codes.

Significant Changes Records

A significant change is a change that was made during the cleansing process that a user may want to review to ensure the data was cleansed as expected.

Currently only a high significance change is identified. An example of a high significant change is when a postcode1 for an address is deleted, or a change in the last name such as "Sanches" was input and output as "Sanchez".

See the data quality statistics table, CLEANSE_CHANGE_INFO_, ([CLEANSE_CHANGE_INFO_ \[page 1356\]](#)) to identify all rows of the input data that were identified as significantly changed.




The reason for the change during the cleanse process is taken into consideration when determining the significance. For example:


- **Reference data:** this includes Global Address Cleanse and Geocoder reference data, and Data Cleanse cleansing package. An example of this reason is the correction of a misspelled city name.
- **Derived:** some elements may be added because they can be derived from the attributes of other elements. For example if the first name is John, a strong given name, a prename of Mr. can be derived.
- **Insufficient confidence:** when there is insufficient confidence in identifying an entity the entire contents may be deleted on output. For example if the input includes an address and the country of the address cannot be confidently identified the address may be deleted on output.

Note

If an entity instance received multiple significant changes, it is counted only once.

Download the task localization file, `dqs_task_localization_<language>.csv` from the SAP Community Network website at <http://scn.sap.com/docs/DOC-68523> to use for translating values in tables.

Column	Data type definition	Description
 OBJECT_KEY	INT	This unique internal ID is assigned at the time of execution for each of the objects created in Data Services. This is a foreign key to the JOB_KEY in the DQVW_AGR_RPTSTATS and DQVW_RE-PORTS_STAT tables.
 OBJECT_ID	NVARCHAR(255)	This ID is unique for each operation within a Data Services work flow. This is a foreign key to the OBJECT_ID in the DQVW_AGR_RPTSTATS and DQVW_RE-PORTS_STAT tables.
 ENTITY_ID	NVARCHAR(12)	Identification number that refers to entities such as person, firm, and address. This is a foreign key to the LOC_ID column in the TASK_LOCALIZATION table where the LOC_TYPE_ID = 1.

Column	Data type definition	Description
 ENTITY_INSTANCE	INT	<p>A number to describe which entity from the record the entry pertains to. For example the ENTITY_INSTANCE for address can be 1-2, and for person 1-6. The ENTITY_INSTANCE corresponds to the instance of the input mapping of an entity, and is therefore dependent on the order of input column mapping.</p>
NUM_RECORDS	INT	This is the count of records input to the cleanse process.
NUM_SUSPECTS	INT	<p>This is the count of entities that have had a change in the data selected to be output that is suspect and it is recommended to review the cleansed output to ensure it is as expected. To identify the suspects the non-summary data quality statistics tables can be used.</p>
NUM_BLANKS	INT	This is the count of entities identified as blank during the cleanse operation. To identify the blanks the non-summary data quality statistics tables can be used.
NUM_HIGH_SIGNIFICANT_CHANGES	INT	<p>This is the count of entities identified as highly significant. To identify the significant the non-summary data quality statistics tables can be used. Note:</p> <div data-bbox="1018 1391 1398 1581"> <p>Note</p> <p>If an entity instance received multiple significant changes, it is counted only once.</p> </div>

❖ Example

Use the DQVW_AGR_RPTSTATS and DQVW_REPORTS_STAT tables to identify the unique key for the results in the data quality statistics tables for the query. The unique key of OBJECT_KEY and OBJECT_ID will be used to analyze the data in the data quality statistic tables. In this example the unique key is:

C.OBJECT_KEY = 412

AND

C.OBJECT_ID = "4f2ba82f-3e7c-44ce-8543-8f5844b44f60"

Sample Code

```
SELECT
    L.DESCRPTION AS "Entity Cleansed",
    C.ENTITY_INSTANCE "AS Entity Instance Cleansed",
    C.NUM_RECORDS AS "Total Number of Rows",
    C.NUM_SUSPECTS AS "Number of Suspects",
    C.NUM_BLANKS AS "Number of Blanks",
    C.NUM_HIGH_SIGNIFICANT_CHANGES AS "Number of Significant Changes"
FROM
    CLEANSE_STATISTICS_ AS C,
    TASK_LOCALIZATION AS L
WHERE
    -- Get the statistics for the desired job and transform.
    C.OBJECT_KEY = 412 AND
    C.OBJECT_ID = "4f2ba82f-3e7c-44ce-8543-8f5844b44f60" AND
    -- Get the description for the ENTITY_ID
    L.LOC_TYPE_ID = 1 AND
    C.ENTITY_ID = L.LOC_ID
ORDER BY
    C.ENTITY_ID,
    C.ENTITY_INSTANCE;
```

RESULTS:

Entity Cleansed	Entity Instance Cleansed	Total Number of Rows	Number of Suspects	Number of Blanks	Number of Significant Changes
Address	1	3000	2021	36	132
Person	1	3000	625	236	296
Organization	1	3000	1560	256	57
Organization	2	3000	152	2205	23
Phone	1	3000	85	13	2
Phone	2	3000	569	1250	22
E-Mail	1	3000	380	22	234
E-Mail	2	3000	546	1552	59

12.3.7 GEOCODE_INFO_CODE_

Provides information about records in your data that were not assigned the highest point of longitude/latitude during the geocoding process.

- Contains a row of statistics for each applicable information code that is assigned during the cleanse process.
- Applicable for Geocoder Cleanse transform

- Contains record-level statistics (non-summary)





Analyze this non-summary table to track the reasons why records were not assigned the highest level of latitude and longitude by using the information code data.


Download the task localization file, `dqs_task_localization_<language>.csv` from the SAP Community Network website at <http://scn.sap.com/docs/DOC-68523> to use for translating values in tables.

For descriptions of information codes see [Information codes \(Data Cleanse\) \[page 827\]](#).

Note

Not all existing information codes apply to data quality statistics.

Column name	Data type definition	Description
 OBJECT_KEY	INT	This unique internal ID is assigned at the time of execution for each of the objects created in Data Services. This is a foreign key to the JOB_KEY in the DQVW_AGR_RPTSTATS and DQVW_RE-PORTS_STAT tables.
 OBJECT_ID	NVARCHAR(255)	This ID is unique for each operation within a Data Services work flow. This is a foreign key to the OBJECT_ID in the DQVW_AGR_RPTSTATS and DQVW_RE-PORTS_STAT tables.
 TABLE_ID	INT	Set to 1.
 ROW_ID	INT	An internal identification generated by the transform that uniquely identifies a row processed by that transform. Appears in each non-summary data quality statistics table and output by the transform. Use this field to join the non-summary data quality statistics tables to the transform output. Subsequently this can be joined to the input source using the input source primary key if it is included in the transform output.
DATA_SOURCE	NVARCHAR(255)	The name of the data source that the record comes from. The value is input from the column input mapped to DATA_SOURCE_ID and is simply passed through to the data quality statistics table.

Column name	Data type definition	Description
 INFO_CODE	VARCHAR(10)	Indicates whether something occurred that prevented the geocode operation from assigning the highest level of latitude/longitude point to the address. Only records that get an information code are written to this table. This is a foreign key to the LOC_ID column in the TASK_LOCALIZATION table where the LOC_TYPE_ID = 11.

❖ Example

Use the DQVW_AGR_RPTSTATS and DQVW_REPORTS_STAT tables to identify the unique key for the results in the data quality statistics tables for the query. The unique key of OBJECT_KEY and OBJECT_ID will be used to analyze the data in the data quality statistic tables. In this example the unique key is:

C.OBJECT_KEY = 412

AND

C.OBJECT_ID = "4f2ba82f-3e7c-44ce-8543-8f5844b44f60"

⇐ Sample Code

```
SELECT
    C.INFO_CODE AS "Info Code",
    L.DESCRPTION AS "Info Code Description",
    COUNT (C.ROW_ID) AS "Count"
FROM
    GEOCODE_INFO_CODES AS C,
    TASK_LOCALIZATION AS L
WHERE
    -- Get the statistics for the desired job and transform.
    C.OBJECT_KEY = 412 AND
    C.OBJECT_ID =
    "4f2ba82f-3e7c-44ce-8543-8f5844b44f60"
    AND
    -- Get the description for the INFO_CODE.
    L.LOC_TYPE_ID = 11 AND
    C.INFO_CODE = L.LOC_ID
GROUP BY
    C.INFO_CODE, L.DESCRPTION
ORDER BY
    C.INFO_CODE;
```

Results:

Info Code	Info Code Description	Count
001	Geocode reference data is not available for this country	64

Info Code	Info Code Description	Count
005	The address entered does not match the Geocode reference data	320
006	The address entered matches multiple addresses in the Geocode reference data and is thus ambiguous	192
008	No entry made for address. This is required in order to match the Geocode reference data	1216

12.3.8 GEOCODE_STATISTICS_



Provides a statistical summary that includes information about the records in your data that were sent through the geocode process.

- Contains a summary row of statistics for the number of total rows and number of rows assigned a longitude/latitude during the geocode process
- Applicable for the Geocoder transform

The table contains summarized data for each Geocode transform to use in providing a high-level insight into the assignment ratio that was output from the geocode process.

Analyze this summary table to identify the assignment success for the highest level of latitude and longitude.

Download the task localization file, `dqs_task_localization_<language>.csv` from the SAP Community Network website at <http://scn.sap.com/docs/DOC-68523> to use for translating values in tables.

Column name	Data type definition	Description
 OBJECT_KEY	INT	This unique internal ID is assigned at the time of execution for each of the objects created in Data Services. This is a foreign key to the JOB_KEY in the DQVW_AGR_RPTSTATS and DQVW_REPORTS_STAT tables.
 OBJECT_ID	NVARCHAR(255)	This ID is unique for each operation within a Data Services work flow. This is a foreign key to the OBJECT_ID in the DQVW_AGR_RPTSTATS and DQVW_REPORTS_STAT tables.
NUM_RECORDS	INT	This is the count of records input to the cleanse process.

Column name	Data type definition	Description
NUM_ASSIGNED	INT	This is the count of records that receive latitude/longitude coordinates. The assignment level is not considered, only that a latitude and longitude is generated.

❖ Example

Use the DQVW_AGR_RPTSTATS and DQVW_REPORTS_STAT tables to identify the unique key for the results in the data quality statistics tables for the query. The unique key of OBJECT_KEY and OBJECT_ID will be used to analyze the data in the data quality statistic tables. In this example the unique key is:

C.OBJECT_KEY = 412

AND

C.OBJECT_ID = "4f2ba82f-3e7c-44ce-8543-8f5844b44f60"

⇐ Sample Code

```
SELECT
  C.NUM_RECORDS AS "Total Number of Rows",
  C.NUM_ASSIGNED AS "Number of Rows Assigned"
FROM
  GEOCODE_STATISTICS_ AS C
WHERE
  -- Get the statistics for the desired job and transform.
  C.OBJECT_KEY = 412 AND
  C.OBJECT_ID = '4b5cf77c-b4a1-479f-acf4-e03b64594a70'
ORDER BY
  C.OBJECT_ID;
```

Results:

Total Number of Rows	Number of Rows Assigned
3392	2816

13 Locales and Multi-byte Functionality

Data Services supports the use of different locales in sources, the Job Server, and targets. It also supports single and multi-byte code pages. By combining these settings, you can control processing across different languages and allow for differences in capitalization, time and date formats, and character sets.

Related Information

[Definitions \[page 1394\]](#)

[Supported locales and encodings \[page 1397\]](#)




13.1 Language packs

Language packs are available for installation, and provide you with locales, other than English (the default locale), for viewing the Data Services user interface and any text that the user interface generates in other languages.

There is no need to reinstall Data Services to acquire a language pack; they can be installed over an existing Data Services installation.

After a language pack is installed, you will be able to select the locale for both the user interface and the displayed data.

- *Product locale*: Specifies the user interface language and all product messages.
- *Preferred viewing locale*: Specifies the locale that the user data should be presented in. For example, date formatting should be presented in the preferred viewing locale.

There are two locations for setting these options: the Locale Selector and the  **Tools**  **Options**  window in the Designer.

Where you make your selections depends on your installation configuration.

13.1.1 Setting locales in the Designer

If you include Designer in your installation, set locals in Designer.

Note

Changing the locale settings in the Designer *Options* window automatically changes the locale settings in the Locale Selector.

1. In the Designer, choose **Tools > Options**.
The *Options* window opens.
2. Expand the *Designer* category, and select *Language*.
3. Select a value for the *Product Locale* and the *Preferred Viewing Locale* options.

13.1.2 Setting locales in the Locale Selector

If you install the engine, but not Designer, set locales using the Locale Selector.

i Note

Changing the locale settings in the Locale Selector will automatically change the locale settings in the Designer's *Options* window.

1. Access the Locale Selector following the instructions below for your operating system:
 - Windows: Choose **Start > Programs > SAP Data Services <x.x> > Data Services Locale Selector**.
 - UNIX/Linux: From the command line, type **./start LocaleSelector.sh**

The Locale Selector window opens.
2. Select a locale for each of the following options:
 - *Product locale*
 - *Viewing locale*
 - *Server Log Locale*
3. Specify the Language, territory, and code page to use for the repository connection and for processing data:
 - To use the default locale for all three, select *Use default database locale*
 - To select locales other than the default, select a locale for *Language and Territory* and a locale for *Code Page*

i Note

UTF-16 is supported as a code page but cannot be selected in the *Locale Selector code page* dropdown list.

13.1.3 Setting locales in UNIX or Linux

Use this procedure to modify the product locales on a UNIX or Linux system.

1. Locate and open your `DSConfig.txt` file. (The default directory is `<$LINK_DIR>/conf/.).`
2. In the `[Locales]` section, change the `ProductLocale` options to the locale you want.
3. Save and close the `DSConfig.txt` file.

13.1.4 Impact of locale settings on Data Services components

The locale settings you choose impact Data Services components differently. Here is a list of those impacts:

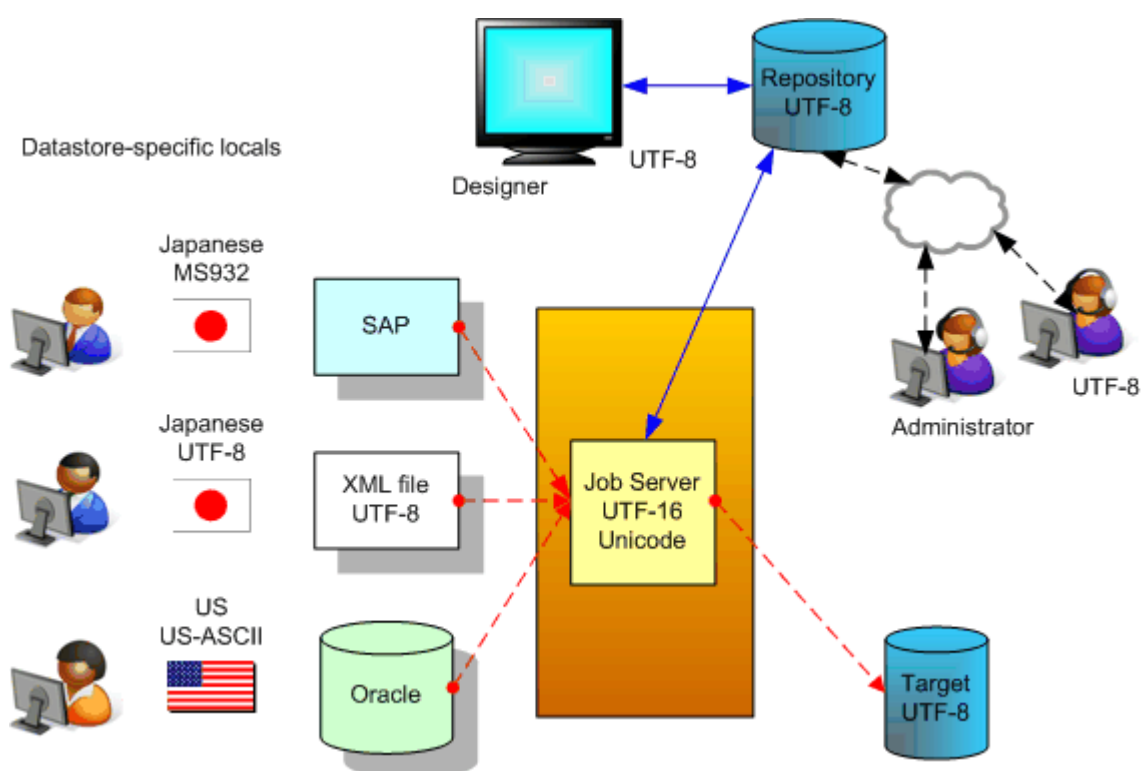
Component	Description
License Manager	License Manager always displays in English.
Management Console	The Management Console's locales are controlled in your internet browser's locale setting.
Documentation	Documentation that is accessed from the Designer will display in the same language as the Designer's <i>Options</i> window <i>Product locale</i> setting.
Log files	Messages written to a log file will be in the language set in the <i>Server Logs locale</i> option in the Locale Selector. This affects only the text of the log files. The date and time-stamps remain the same.

13.2 Locale support

By supporting different locales, Data Services allows you to configure an enterprise environment in which components process data in different human languages and then load the data to a target data code page.

For example, you can configure locales for the following sources:

Language	Territory	Code page
Japanese	Japan	shift_jis
Japanese	Japan	UTF-8
English	United States	US-ASCII



To use the locals that Data Services supports, set locales for each of the following areas:

- Database (source, target, and repository) or application (SAP, PeopleSoft, Oracle, Siebel, JDE).
- Database client.
- Datastore connection to a database or application.
Datastore locales must match the locales of source and target database clients. This allows the datastore to move data between Data Services and each database without possible data corruption. If the database locale differs from its database client locale, it is your responsibility to ensure that the database transcodes the data before it reaches or after it leaves Data Services.

i Note

To avoid the necessity of setting locales for a database client and a Data Services datastore, you can process with or without UTF-16 Unicode.

i Note

All adapter datastores are automatically set by Data Services to the code page UTF-8. They are handled the same way as XML message sources and targets.

- File format (flat file, XML Schema, and DTD): Match the file format locale to that of each source or target file.
- Job Server: Data Services uses the Job Server's locale for the engines it spawns. The Designer also uses the Job Server's locale as its repository connection locale. To avoid possible data corruption, in case the SAP Sybase repository or DB2 repository database codepage is not UTF-8, use the same locale settings for the repository, its client, and the Job Server.

Related Information

[Processing with and without UTF-16 Unicode \[page 1384\]](#)

[File format locales \[page 1388\]](#)

13.2.1 Locale selection

Data Services automatically sets the locale for the Job Server, Designer, and Management console.

13.2.1.1 Job Server locale

The Job Server locale is set after installation to `<default>`. This default Job Server locale enables Data Services to automatically set the locale for the repository connection (for the Designer) and to process job data (for the Job Server) according to the locale of the datastore or operating system. This capability enables Data Services to automatically change the locale for better performance (for example, set the locale to non-UTF-8 if the datastore is non-unicode data).

The following table shows different datastores and Job Server locale settings and the locale that Data Services automatically sets for the data flow. In this table, the Job Server locale is set to `<default>` and derives its value from the operating system.

Datastore locale 1	Datastore locale 2	Job Server locale	Data flow locale
Single-byte code page	Multi-byte code page	Single-byte or Multi-byte code page	Unicode
Multi-byte code page	Multi-byte code page	Single-byte code page	Unicode
Multi-byte code page	Multi-byte code page	Multi-byte code page	Unicode
Single-byte code page 1	Single-byte code page 2	Single-byte code page 3	Unicode
Single-byte code page 1	Single-byte code page 2	Multi-byte code page	Unicode
Single-byte code page 3	Single-byte code page 3	Multi-byte code page	Unicode
Single-byte code page 3	Single-byte code page 3	Single-byte code page 1	Single-byte code page 3

The following table summarizes the locale that Data Services sets for each data flow when the locale of the Job Server is set to `<default>`. Different data flows in the same job can run in either single-byte or Unicode.

Datastore locales in data flow	Job Server locale	Locale that Data Services sets
One datastore has multi-byte locale	Single-byte or multi-byte	Unicode
Different single-byte locales	Single-byte or multi-byte	Unicode
Same single-byte locale	Multi-byte	Unicode
Same single-byte locale	Single-byte	Single-byte

Related Information

[Locales and Multi-byte Functionality \[page 1378\]](#)

[Locale selection \[page 1382\]](#)

13.2.1.2 Designer locale

The Designer uses the Job Server locale to move data between the Designer and the repository.

The Designer expects to receive data from the repository in the Job Server's locale. Objects you create in the Designer are represented internally using a textual language (ATL) that is sent to and received from the repository's database client in the form of SQL statements. Use the same locale when installing the Job Server as you set for your repository and its client to support Data Services' internal language.

The Designer also has its own locale which is automatically set to that of its Microsoft Windows operating system locale. The Designer automatically transcodes input data from its locale to the Job Server locale when it interacts with the repository.

Related Information

[Locales and Multi-byte Functionality \[page 1378\]](#)

[Locale selection \[page 1382\]](#)

13.2.1.3 Management Console locale

The Management Console's locale is automatically set to UTF-8. By using UTF-8 (a Unicode encoding that supports all languages), Data Services ensures data integrity in the Management Console. All Data Services logs (error, trace, and monitor) are generated by the engine in UTF-8. When the Designer reads logs, it transcodes their content from UTF-8 to the Designer locale.

Related Information

[Locales and Multi-byte Functionality \[page 1378\]](#)

[Locale selection \[page 1382\]](#)

13.2.1.4 Overriding the default Job Server locale

You can override the default locale for the Job Server by using the Data Services Locale Selector utility.

Choose ► *Start* ► *Programs* ► *SAP Data Services <x.x>* ► *Data Services Locale Selector* ►.

For more information, see “Guidelines for setting locales”.

13.2.2 Code page support

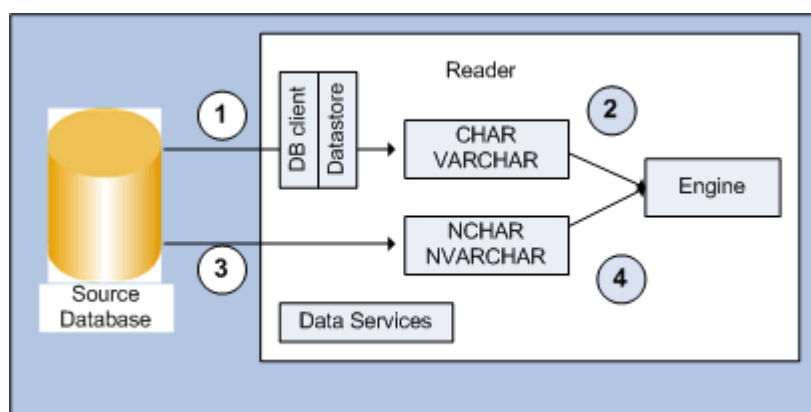
The code page you select in each locale used by a job determines whether transcoding occurs during job processing. Data Services automatically transcodes between different code pages when necessary to support complex, multi-language data management in a single job.

However, it is your responsibility to set up the connections between Data Services and its repository, sources, and targets to avoid data corruption due to mismatched locale settings between Data Services and its connections to external systems. Therefore, ensure you use the same code page as the database middleware to which you are connecting.

13.2.2.1 Processing with and without UTF-16 Unicode

The Job Server supports reading and loading national character-set data types (`nchar`, `nvarchar`, `nvarchar2`, `graphic` and `var-graphic`). Data Services automatically handles these data types using UTF-16. If extracting or loading data that uses only these data types, you do not have to set locales for a database client and its datastore. If the data is multi-byte, the Job Server locale should be set to a multi-byte code page, such as `UTF-8` or `shift_jis`.

For example, here is the path that your data takes from a source database to a Data Services engine when data is read during a job run.



The upper arrow shows a normal job run and the locales the job uses to support data integrity.

1. Data Services assumes the database transcodes data into the database client code page, as needed, before it uses its datastore code page to read data from the database. The client and datastore code pages must match otherwise Data Services cannot recognize the format of the data. Note that the data types are not national character-set data types.
2. If the datastore code page is different from the Job Server code page, Data Services transcodes it then processes the job. Note that the client and datastore are not in the path.
The lower arrow shows a national character-set data type job run:
3. Data Services reads the data formatted for national character-set data types using a UTF-16 code page without using client and datastore code pages.
4. The engine transcodes the data from UTF-16 to the Job Server's code page before it is processed.

When data continues to a target database, the processes are reversed. Data Services automatically transcodes the data in the national character-set data type path back into UTF-16 before loading it into a target. In a normal job run path, if the datastore code page differs from the Job Server code page, the engine transcodes the data to the target's code page before it is passed on to the database client. The datastore locale and the database client locale must match so that data is accurately sent into the database where it might again need to be transcoded into a different locale for storage.

Data Services support of national character-set data types is restricted to the specific source and target databases. For example:

- Oracle with `nchar` or `nvarchar2`
- Microsoft SQL Server with `nchar` or `nvarchar`
- DB2 with `graphic` or `vargraphic`

Data Services can also extract, transform, and load a single table with both national character-set data types and other data types. In this case, the data in the columns with the national character-set data types uses the UTF-16 path and the other data uses the datastore path through Data Services.

In the Designer, you can also assign Unicode as a code page by creating a Microsoft SQL Server datastore connection and selecting [UTF16](#) or [UTF8](#) as its code page.

National character-set data types help you avoid having to set locales for connections to database and application sources and targets. However, you still must set locales for the Job Server and for file formats (if you use files in your job).

Also, when Data Services imports a table with columns using any character data type (`nchar`, `nvarchar`, `varchar`, `char`, and so on), it imports the column size in number of characters (not bytes). Similarly, while creating a new column in Query objects, Data Services assumes the column size is in number of characters. As the number of bytes per character varies from code page to code page, at runtime, the Data Services engine allocates memory based on the Job Server's code page.

Related Information

[Column sizing \[page 1393\]](#)

13.2.2.2 Minimizing transcoding

As a rule, transcoding impacts job performance.

- Use the same locale for all components and use a single-byte code page if possible.
- If a datastore or file format and the Job Server use a different locale, Data Services automatically transcodes the data, which supports a multi-language enterprise environment.

Data Services minimizes the impact of transcoding for equivalent code sets such that transcoding between the following code page pairs does not impact performance.

Superset	Subset
cp1252	ISO88591 (LATIN1)
cp1250	ISO88592
cp1251	ISO88595
cp1253	ISO88597
cp1254	ISO88599
cp1255	ISO88598
cp1256	ISO88596
cp1257	ISO88594

Related Information

[Supported locales and encodings \[page 1397\]](#)

13.2.3 Guidelines for setting locales

13.2.3.1 Job Server locale

The Job Server locale is used by the engines it spawns as well as the Designer's repository connection locale.

If the locale of the Job Server is set to `<default>` after installation, the Job Server takes its locale from the operating system of the host computer where it is installed. If you process multi-byte data with the operating system locale set to single-byte, set the Job Server's locale to the same code page or a superset of the data code page to avoid data corruption. For more information, see Example 3 in the "Example locale settings" section.

If your jobs will run in a multi-language environment, Data Services automatically sets the Job Server's locale to a superset of all datastore and file format locales.

If you do not have a multi-language environment, use a single-byte code page and use the same settings for all locale values or use only locales with code pages that minimize transcoding. This strategy ensures the best performance.

You can override the default locale for the Job Server by using the Data Services Locale Selector utility. Choose **Start > Programs > SAP Data Services <x.x> > Data Services Locale Selector**.

Also make sure that the locale of the repository's database client (installed on the Designer's computer) matches the Job Server locale. The Designer uses the Job Server's locale to ensure that it passes accurate data to the repository.

Related Information

[Minimizing transcoding \[page 1386\]](#)

[Supported locales and encodings \[page 1397\]](#)

[Example locale settings \[page 1390\]](#)

13.2.3.2 Database, database client, and datastore locales

Set a database and its database client locales using your database software.

Set a Data Services datastore to the same locale as the application or database client to which it connects. Data Services automatically sets each datastore locale to `<default>` in order to match that of the Job Server. However, if your sources or targets use different locales, manually modify the [Language](#) and [Code page](#) options under the [Advanced](#) button in the Datastore Editor.

When you view table data with the View Data feature of the Designer, Data Services formats the values of numeric data type columns according to the number format of its locale territory. For example, suppose a datastore that is connected to an Oracle database with a datastore locale of `deu_de.cp1252`. With the Data Services locale set to `eng_us.cp1252`, View Data will display numeric values with a dot (.) as the decimal separator.

When reading and loading numeric data from databases, Data Services automatically determines the number format appropriate for each database, which does not depend on the Job Server locale territory. However, if the datastore table contains numeric values in string data type columns and an implicit conversion from string to numeric data type is required, Data Services expects that the number format matches the format of its locale territory.

If the format of numeric values in string data type columns does not match the Job Server locale territory, use the `to_decimal_ext` function or the `to_decimal` function to convert the string to a numeric data type by specifying the correct thousand and decimal separators. Similarly, when loading numeric values to string-type columns in a datastore, Data Services formats numbers according to the Job Server locale territory format. If you need to convert the data to a number format used by a different territory, use the `to_char` function.

Locales apply for all profiles created from each datastore.

All adapter datastores are automatically set by Data Services to the code page UTF-8. They are handled the same way as XML message sources and targets.

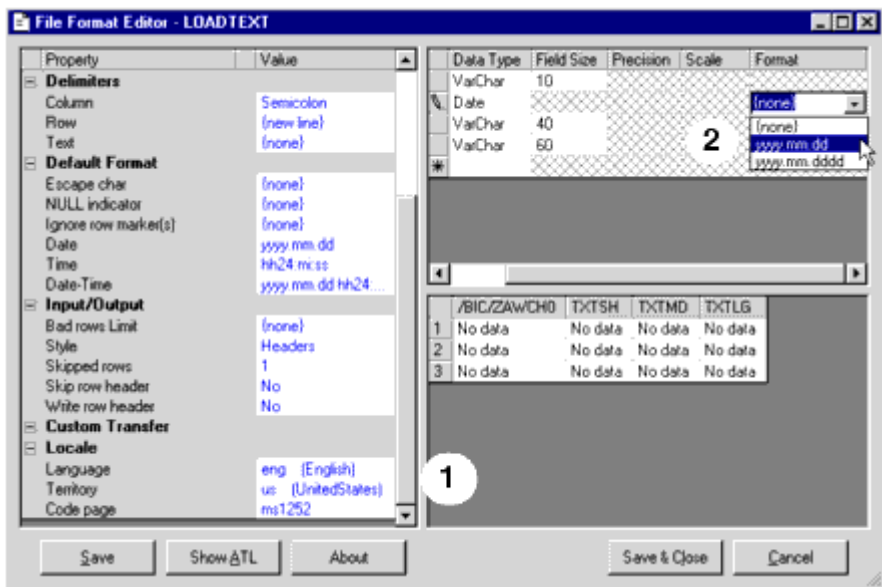
Related Information

- [Datastore \[page 56\]](#)
- [XML encodings \[page 1389\]](#)
- [to_char \[page 1221\]](#)
- [to_decimal \[page 1225\]](#)
- [to_decimal_ext \[page 1226\]](#)

13.2.3.3 File format locales

For flat files, the <default> locale is automatically set to match that of the Job Server. However, you can use the [Locale](#) section on the file format editor to specify the language and code page that corresponds to the file data.

Refer to the following image:



1. As in datastore locales, the [Territory](#) option for a flat file's locale is not active in Data Services.
2. However, you can change the data type for a flat file column and then enter the format that you want to use for a date or any numeric data type using the [Column Attributes](#) work area.

Related Information

- [File format \[page 147\]](#)

13.2.3.4 XML encodings

For XML Schemas or DTDs used as sources, the default encoding (code page) for messages is assumed by Data Services to be UTF-8. Header information is ignored. Data Services transcodes inbound data to the Job Server code page (if necessary) before processing it. For XML files Data Services reads header information at run-time. If encoding information is not specified, Data Services assumes it is UTF-8. To edit the encoding for XML file sources, do so in the original file then rerun the job.

For XML Schemas or DTDs used as targets, the default encoding is automatically set to UTF-8.

- If you use an XML Schema or DTD as a file, you can change the encoding using the target editor.
- If you use an XML Schema or DTD as a message, the encoding cannot be changed. Data Services transcodes out-bound message configuration information (like global variable data) into UTF-8.

Related Information

[Target XML files, messages, and templates \[page 271\]](#)

13.2.3.5 Overriding the database client locale

By default, most database clients set their locale to the same locale used by the operating system. However, you can override the default setting by using either an environment variable or client configuration tool. We recommend using the methods below to override the client locale for different database types.

Database	Default setting	Environment variable	Driver setting
DB2	OS Locale	DB2CODEPAGE	N/A
Informix	OS Locale	CLIENT_LOCALE	Client Locale field
MySQL	Latin1	N/A	Character Set field
Netezza	N/A	N/A	ASCII (8-bit driver) UTF16 (Unicode driver)
Oracle	OS Locale	NLS_LANG	NLS_LANG
SAP ASE	OS Locale	LC_ALL	N/A
SAP Sybase IQ	OS Locale	N/A	Character Set field
Teradata	ASCII	N/A	Session Character Set field

i Note

For Microsoft SQL Server databases, the only way to change the database client locale is to change the operating system locale. For generic ODBC databases, see your ODBC driver configuration documentation.

13.2.3.6 Example locale settings

The following examples illustrate setting up locales under different circumstances.

13.2.3.6.1 Example 1: Windows system set to Japanese

Suppose you are running Data Services on a Windows system that is set to Japanese and want to process Japanese data, with the Data Services code page set to default, `<default>_<default>.<default>`. All datastore and file formats are also set to the `<default>` code page.

In this situation, the Job Server takes its locale from the server's system locale, `jpn_jp.shift_jis`. All datastore and file formats get the `shift_jis` codepage.

To access the data correctly, the database client must also be configured for a locale equivalent to `jpn_jp.shift_jis`. Refer to the database client documentation for information on how to obtain and configure the correct locale name.

Related Information

[Locale support \[page 1380\]](#)

[Code page support \[page 1384\]](#)

[Example locale settings \[page 1390\]](#)

13.2.3.6.2 Example 2: Windows system set to English

Suppose you are running Data Services on a Windows system that is set to English (`ms1252`) and want to process Japanese data with the datastore code page set to the `<default>`. You must set the Job Server's code page to Japanese (`shift_jis`) to avoid data corruption.

Related Information

[Locale support \[page 1380\]](#)

[Code page support \[page 1384\]](#)

[Example locale settings \[page 1390\]](#)

13.2.3.6.3 Example 3: Multiple databases and different character sets

Suppose you are running Data Services on a Windows system and want to extract data from multiple databases using different character sets, such as `shift_jis`, `latin1`, and `latin9`, and then load that data to a single UTF-8 target database.

To support this scenario, set the Data Services code page to `eng_US.utf8`, and configure each datastore locale to match the locale of the linked database.

! Restriction

Because the database client code page must match the database server code page, there can be only one client character set per system. This restriction does not apply to databases that allow you to configure the character set via an ODBC driver setting.

Related Information

[Locale support \[page 1380\]](#)

[Code page support \[page 1384\]](#)

[Example locale settings \[page 1390\]](#)

13.3 Multi-byte support

Data Services supports various multi-byte code pages that are typically specific to each language. Data Services also provides support for two Unicode encodings: UTF-8 and UTF-16. These are multi-byte code pages that support most of the world's languages.

13.3.1 Multi-byte string functions

All Data Services string functions support multi-byte data.

For example, when using the functions `INDEX`, `LENGTH`, `RPAD`, and `SUBSTR`, the sizes and offsets used as arguments or return values are expressed as number of characters, not number of bytes. You can also see this logical, intuitive behavior when these functions are pushed down and evaluated by the database. Similarly, when Data Services evaluates the SQL predicate `LIKE`, the single-character wildcard `"_"` matches exactly one character, not one byte.

13.3.2 Numeric data types: assigning constant values

Use care when assigning a constant to a numeric variable or column in Data Services.

Assigning a value as a numeric directly

If a numeric value is not within quotes—for example, `$comm = 123.45`—then Data Services stores the value as a numeric. If the value is stored as numeric, then dot (".") is the only recognized decimal separator in the Designer, regardless of the locale. While executing the job, however, the Data Services engine automatically converts the value to appropriate decimal separator for the locale.

Syntax	Comment
<code>\$VALUE = 123.45</code>	Correct syntax.
<code>\$VALUE = 123,45</code>	Wrong syntax. Use a comma "," as a decimal separator inside Data Services only if the operating system's decimal separator is a comma "," and the value is within quotes.

Assigning a value in string format

If a numeric value is stored within quotes—for example, `$comm = '123,45'`—then the Designer stores the value in a string format and while executing the job, the Data Services engine automatically converts the value from the string to the appropriate numeric data type. When the Data Services engine converts a string to a numeric format, it uses the decimal separator for the specified Job Server locale.

For example, if the Job Server locale is set to German regional settings (`deu_de.cp1252`), then Data Services uses a comma (",") as the decimal separator when converting a value from a string to appropriate numeric data type.

Syntax	Comment
<code>\$VALUE = 123,45</code>	Correct syntax.
<code>\$VALUE = 123.45</code>	Wrong syntax. If the operating system's decimal separator is a comma "," and the value is within quotes, then Data Services tries to process a dot "." as a thousands separator.

If the Job Server locale is set to English regional settings (`eng_us.cp1252`), then Data Services uses a dot (".") as the decimal separator when converting a value from a string to an appropriate numeric data type.

Syntax	Comment
<code>\$VALUE = 123.45</code>	Correct syntax.

Syntax	Comment
<code>\$VALUE = 123,45</code>	Wrong syntax. If the operating system's decimal separator is a dot "." and the value is within quotes, then Data Services tries to process a comma "," as a thousands separator.

Note

Using the incorrect decimal separator can generate incorrect results. Similarly, having a thousands separator can generate incorrect results. It is recommended that you do not use a thousands separators when converting a value from string to numeric.

13.3.3 Byte Order Mark characters

The Unicode standard includes the use of Byte Order Mark (BOM) characters as a signature for file data.

Data Services supports BOM characters. When it reads data from a file, the Data Services engine trims BOM characters. Data Services supports the following BOM characters and their corresponding encodings:

BOM Characters (Bytes)	Encoding Form
FE FF	UTF-16 big-endian
FF FE	UTF-16 little-endian
EF BB BF	UTF-8

13.3.4 Round-trip conversion

While transcoding, if Data Services encounters round-trip conversion conflicts, it transcodes to the first code point match (ascending order of the hexadecimal values) in the target code page. For example, consider two different Japanese characters that are mapped to different hexadecimal code points in the shift_jis code page (EEFA and FFE4) but then are mapped to the same UTF-16 code point (FFE4). In this case, when transcoding back from UTF-16 to shift_jis, Data Services converts them both to code point EEFA.

13.3.5 Column sizing

The number of bytes per character can vary from one code page to another. For example, the "ᄀ" which represents a special "A" in the Korean ksc-5601 code page, needs 2 bytes to represent the character, while the UTF-8 code page needs 3 bytes to represent the same character.

Code page	Value	Hex values	Bytes
ksc-5601	special A	A3 C1	2
UTF-8	special A	EF BC A1	3

Data Services represents sizes in number of characters. Internally, Data Services allocates enough memory to store multi-byte characters.

If the datastore code page is different from the Job Server code page, then transcoding must occur, which may result in the need for extra space allocation.

13.4 Limitations of multi-byte support

There are several limitations to SAP Data Services' support for multi-byte characters:

- The software supports a variety of single- and multi-byte code pages, but it does not support UCS-4, or SAP application blended code pages. In addition, the software does not support Surrogate Pairs. EBCDIC is supported for COBOL files only.
The software does not support EBCDIC code pages for datastores as they are not ASCII- or UTF-8 - compatible. This is not a problem for users targeting or sourcing data from IBM systems if the engine is running on a non-IBM platform. The IBM system will transparently convert its data to/from EBCDIC when communicating with a foreign architecture.
- Each data flow can process in different locales. However, if you need to change the default locale setting, you need to use a different installation of the Job Server on a different computer, because the automatic locale setting will be disabled.
- The software does not yet fully address all formatting issues. For instance, it supports the "dot" and "comma" currency formats used for most European currencies, but does not support the "tick" and "space" currency format used in Switzerland.

13.5 Definitions

With regard to locales and multi-byte functionality associated with Data Services, refer to the following terms:

Locale and multi-byte functionality terminology

Term	Definition
Locale	<p>Consists of three values related to world regions that control the format of data when it is stored, processed, or displayed. To specify a locale for the Job Server, you must select a Language, Territory, and Code page value.</p> <p>Datastore and file format locales do not require that you set the Territory value for a locale.</p> <p>Database sources and targets might not need locale values specified.</p>
Language	Specifies the locale value for a human language.
Territory	<p>Specifies the locale value for a geographical location (usually the country) where a locale language is used. The pairing of a language with a territory determines factors such as date format, time format, decimal separator, currency format, and so on. Data Services uses territory values to process the following data types:</p> <ul style="list-style-type: none"> • date • datetime • decimal • double • int • interval • numeric • real • time • timestamp
Code page	<p>A table of characters that associates each character with a numeric index (code point value). Data Services uses a code page value to transcode varchar data types. Most languages require their own code pages, although some code pages can represent multiple languages. Most code pages are compatible with <code>US-ASCII</code> for code points below 128.</p> <p>This means for example, that the Japanese code page <code>shift_jis</code> also partially supports other languages such as English. However, use <code>Japanese</code> as the Language value, <code>Japan</code> as the Territory value, and <code>shift_jis</code> as the Code page to avoid possible data corruption in a Japanese locale. It is your responsibility to select corresponding values for locales. While Data Services validates that locale values are entered, it does not validate that they are realistic.</p>
Single-byte	An encoding or code page in which each character is represented by one byte.

Term	Definition
Multi-byte	An encoding or code page in which each character is represented by one or more bytes. Some languages, like Korean and Chinese, can only be represented by multi-byte characters. Use multi-byte code pages to process characters for those languages.
Encoding	The process of representing a code page character as one byte (single-byte encoding) or a sequence of bytes (multi-byte encoding).
Transcode	<p>Converts data from one code page to another.</p> <p>To support ETL environments in which sources with different locales are processed in the same job, Data Services supports transcoding. Note that transcoding can impact performance.</p>
Unicode	Provides a unique number for every character and a method to implement ISO/ISE 10646. Data Services supports UTF-8 and UTF-16 Unicode transformation formats.
UTF-8	Serializes a Unicode code point as a sequence of one to four bytes depending on the complexity of the character (single-byte characters use one byte and multi-byte characters use up to four). Data Services allows you to select UTF-8 as a code page for the Job Server and connections to sources and targets.
UTF-16	<p>In Data Services, standardizes each Unicode code point at two bytes for each character. Allows access to 63k characters as 16-bit units.</p> <p>Data Services supports UTF-16 for:</p> <ul style="list-style-type: none"> • A Microsoft SQL Server database when its datastore code page is set to <code>utf8</code> or <code>utf16</code> • National character-set data types in the following databases: <ul style="list-style-type: none"> ◦ Oracle with <code>nchar</code> or <code>nvarchar2</code> ◦ Microsoft SQL Server with <code>nchar</code> or <code>nvarchar</code> ◦ DB2 with <code>graphic</code> or <code>vargraphic</code> <p>When using UTF-16 support, you do not have to set locales for connections to database sources and targets.</p>

Related Information

[Locales and Multi-byte Functionality \[page 1378\]](#)

13.6 Supported locales and encodings

For a language, territory, code page or encoding you can also select `<default>`. For a Job Server locale, this means that the value is read from the operating system's locale. For example, you log in to your Windows session as `Davis` with a user locale of `eng_gb.cp1252`. When you install a Job Server, it picks up the same locale and displays it as the default (`eng_gb.cp1252`). At this point you can:

- Accept these values. The Job Server will always use the `eng_gb.cp1252` locale.
- Use `default` for one or all values, for example by entering `<default>_<default>.<default>`. The Job Server's locale will always match the operating system's locale.
- Edit this locale to, for example `eng_us.cp1252`. The Job Server will always use the `eng_us.cp1252` locale.

Your choice will depend on how you want to set up your system.

Note

The combination of language code `zh` and territory `cn` maps to Simplified Chinese, while the combination of `zh` and `tw` maps to Traditional Chinese.

Supported languages

SAP Data Services supports all three-letter language abbreviations specified in the ISO 639-2/T standard.

Supported territories

SAP Data Services supports all two-letter territory abbreviations specified in the ISO 3166-1 standard.

Supported code pages

SAP Data Services supports the following code pages:


Code page	Description		
	XML Encoding	Multi-byte	Unicode Ver.
big5	Traditional Chinese Big 5 plus Microsoft extensions. User-defined range added to match Windows 2000 SP4.		
	big5	Yes	Microsoft code page 950 3.0
big5-hkscs	Traditional Chinese Big 5 plus Hong Kong Supplementary Character Set.		
	big5-hkscs	Yes	Microsoft code page 950 3.0

Code page	Description			Unicode Ver.
	XML Encoding	Multi-byte	Vendor	
cp850	Latin-1 (Western European languages)			2.0
	WINDOWS-850		Microsoft code page 850	
cp1250	Latin-2 (Central Europe)			2.1
	WINDOWS-1250		Microsoft code page 1250	
cp1251	Cyrillic (Slavic)			2.1
	WINDOWS-1251		Microsoft code page 1251	
cp1252	Latin-1 (ANSI), ISO 8859-1 plus Microsoft extensions			2.1
	WINDOWS-1252		Microsoft code page 1252	
cp1253	Greek			2.1
	WINDOWS-1253		Microsoft code page 1253	
cp1254	Latin-5 (Turkish), ISO 8859-9 plus Microsoft extensions			2.1
	WINDOWS-1254		Microsoft code page 1254	
cp1255	Hebrew			2.1
	WINDOWS-1255		Microsoft code page 1255	
cp1256	Arabic			2.1
	WINDOWS-1256		Microsoft code page 1256	
cp1257	Baltic Rim			2.1
	WINDOWS-1257		Microsoft code page 1257	
cp1258	Vietnamese			2.1
	WINDOWS-1258		Microsoft code page 1258	
cp936	Simplified Chinese, GB 2312-80 plus Microsoft extensions. User-defined range added to match Windows 2000 SP4.			3.0
	iso2022cn	Yes	Microsoft code page 936	
euc-jp	Japanese Extended UNIX Code (incl. JIS X 0212)			2.1
	EUC-JP	Yes	Japanese EUC (JIS X 0201-1976, JIS X 0208-1990, JIS X 0212-1990)	
euctw	Traditional Chinese (Taiwan) Extended UNIX Code			
	EUC-TW	Yes		
gb18030	Chinese National Standard (supports both simplified and traditional Chinese characters)			
	GP18030	Yes		
ibm-874_ p100-1995	Thai			
		Yes		
iso-8859-1	Western European			2.1
	ISO-8859-1		ISO/IEC 8859-1:1987	

Code page	Description			
	XML Encoding	Multi-byte	Vendor	Unicode Ver.
iso-8859-15	Latin-9, Western European			
	ISO-8859-15		ISO/IEC 8859-15:1999	2.1
iso-8859-2	Latin-2, Eastern European			
	ISO-8859-2		ISO/IEC 8859-2:1987	2.1
iso-8859-3	Latin-3, Southeast European			
	ISO-8859-3		ISO/IEC 8859-3:1988	2.1
iso-8859-4	Latin-4, Baltic			
	ISO-8859-4		ISO/IEC 8859-4:1988	2.1
iso-8859-5	Cyrillic			
	ISO-8859-5		ISO/IEC 8859-5:1988	2.1
iso-8859-6	Arabic			
	ISO-8859-6			
iso-8859-7	Greek			
	ISO-8859-7		ISO/IEC 8859-7:1987	2.1
iso-8859-8	Hebrew			
	ISO-8859-8		ISO/IEC 8859-8:1988	2.1
iso-8859-9	Latin-5, Turkish			
	ISO-8859-9		ISO/IEC 8859-9:1989	2.1
ksc-5601	Korean KS C 5601-1992 plus Microsoft extensions. Currency mapping changed and user-defined range added to match Windows 2000 SP4.			
	KSC_5601	Yes	Microsoft code page 949	3.0
shift_jis	"Standard" Japanese Shift-JIS without Microsoft extensions			
	Shift_JIS	Yes	Shift-JIS (JIS X 0201-1976, JIS X 0208-1990)	2.1
us-ascii	7-bit ASCII			
	ISO-8859-1		ISO/IEC 646	
utf-8	UTF-8 encoding of Unicode			
	UTF-8	Yes		2.1

14 Python

SAP Data Services supports the Python programming language for writing expressions with the User-Defined and Match transforms. In this section, you can find explanations of Python methods and some examples.

If you want more information about the Python language, see the Python help file, which is installed with the software. By default, it is located at `<LINK_DIR>\DataQuality\python\doc\python25.chm`. The Python web site, at www.python.org , also contains valuable information.

14.1 About Python

Python is an open-source, object-oriented scripting language. Python is installed with your SAP Data Services installation; therefore, you are equipped with everything you need to begin coding. The software even has its own Python Expression editor, which is accessed from the Match and User-Defined transforms.

By using Python with the software, you can customize transforms to meet your specific needs during processing.

14.1.1 Python module

The software has its own Python module that contains five classes:

- FLDataCollection class
- FLDataManager class
- FLDataRecord class
- FLProperties class
- FLPythonString class

Each of these classes has one or more methods.

14.1.2 Supported transform

You can use the Python module with these transforms:

- Match
- User-Defined

14.1.3 Third-party Python libraries

To ensure that your Python expressions run correctly, make sure that all third-party python libraries are in the appropriate dynamic library path for your operating system so that the dependencies are resolved. If you find that a Python library is not working correctly, update the library path (LD_LIBRARY_PATH for Solaris and Linux, LIBPATH for AIX, and SHLIB_PATH for HP) in the environment where the AL_JobService is installed, and restart the job service.

14.1.4 Processing mode

The User-Defined transform can run in two processing modes. You specify the mode in the User-Defined Editor. The processing mode determines how the Python expression is applied:

- *Per Record*: Applies the expression to each record. This method is useful for formatting your data, such as making the data all uppercase. You cannot add new records into the data flow with this option.
- *Per Collection*: Applies the expression to the entire data collection. For example, the software could go through each record in a collection to decide if a certain statement is true and then take an action on the entire collection. Use this option when adding or deleting new or duplicate records.

14.1.5 Internal coding

For use with Per Record processing mode, much of the Python coding is done internally, so that you don't need to worry about properly importing the Python module. Most of the examples in this section do not include any import syntax.

14.1.6 Unicode in Python

The software is Unicode-enabled. Therefore, all Python methods require Unicode input and all return values are in Unicode. In Python syntax, you must alert Python that you are processing Unicode data, for example (where "u" indicates Unicode):

```
record.SetField(u'NAME', u'value')
```

⚠ Caution

Make sure to use a "u" to indicate Unicode every time you use a Unicode string to look up field names; for example, in a GetField, SetField, or SendToPipe method. If you do not, an error or crash may occur.

14.1.7 Clean up new memory references

Some of the Python methods return new objects when you call the method. You can see which methods return new objects by reviewing the “Return value” heading under each method's section in this document. Any method with a return value returns new objects.

Whenever you create a new object (such as a variable) that refers to one of these methods, you are also creating a new memory reference.

Caution

Be sure to clean new memory references. If you don't clean up these references in your scripts, you may notice a memory leak after running projects where your script is present.

For example, assume that you want to retrieve the value of a field using the `GetField()` method, and save that value as a variable. Because that method returns a value, you must clean up that reference by deleting the variable at the end of the script. For example:

```
Master = SRC.GetField(u'input.code')
del Master
```

However, if you use the `NewDataRecord()` method and do not use `Collection.AddRecord(newRec)`, use the `DeleteDataRecord()` method before you use the `del` command. Otherwise, the memory allocation remains. For example:

```
newRec = DataManager.NewDataRecord(1)
DataManager.DeleteDataRecord(newRec)
del newRec
```

If you use `Collection.AddRecord(newRec)`, you do not need to use the `DeleteDataRecord()` method before you use the `del` command, because you do not own the record. For example:

```
Collection.AddRecord(newRec)
del newRec
```

The following example gets the first record from a collection and deletes it.

```
newRec = DataManager.NewDataRecord(0)
Collection.GetRecord(newRec, 1)
Collection.DeleteRecord(newRec)
del newRec
```

14.1.8 Using Mapped_Name

When you set up fields for the transforms that use Python scripts in Per Record mode, you can specify the `Mapped_Name`. The value of this option indicates an “alias” so that you can refer to your field easily in your Python code.

In some transforms, you can specify a `Mapped_Name` for fields on both input and output. In this case, be sure to use unique values for these options. If you use the same value on both input and output, you will receive an error.

When you use the `GetField` and `SetField` methods, make sure the you enter the `Mapped_Name` correctly in the Python code and map the input field in the transform. If the `Mapped_Name` is not used correctly and mapped, you may encounter the following error:

```
FlDataRecord::GetField() error: Invalid field name MAPPED_RECNO.
```

14.1.9 Using paths with substitution parameters and custom options

Be cautious when writing any sort of expression that might reference a path. The backslash (`\`) symbol is used in Python to indicate an escaped character. For example, `"\u"` indicates Unicode. Therefore, if you use certain paths in your expression, it won't be read properly because the escaped character will be recognized as such. For example:

```
c:\userdata\myfolder
```

In this specific instance, you would likely receive the following error when you run the project that contains this User-Defined transform: `"UnicodeError: Unicode-Escape decoding error: truncated \uXXXX escape"`.

It is also important to consider this when using substitution parameters or custom option in your expression. If your substitution parameter refers to a path, it may encounter the same issue.

You can avoid this issue in several different ways:

- Use forward slashes.

```
u"c:/userdata/myfolder/test_file.txt"
```

- Use a double backslash in the path.

```
u"c:\\userdata\\myfolder\\test_file.txt"
```

- Enclose the substitution value with an `"r"`, which indicates a raw string.

```
r'[$$TEST]'
```

14.2 Creating an expression with the Python Expression editor

The Python Expression editor, which is similar to the Smart editor, helps you create your Python expressions. The editor provides basic programming features such as keyword highlighting, auto-completion, auto-indentation, and code tool tips.

Related Information

[Smart editor \[page 299\]](#)

[Opening the Python Expression editor from the User-Defined transform \[page 1404\]](#)

[Writing Python code \[page 1405\]](#)

[Validating syntax \[page 1406\]](#)

[Fixing syntax \[page 1406\]](#)

[Find and Replace \[page 1406\]](#)

14.2.1 Opening the Python Expression editor from the User-Defined transform

Before you use the Python Expression editor, define your input and output fields for the User-Defined transform in the transform editor.

1. In the User-Defined transform editor *Options* tab, click the *Edit Options* button to open the User-Defined Editor.

You can also select the User-Defined transform in the data flow and choose ► *Tools* ► *User-Defined Editor* ►.

2. Decide which processing mode you want to use and select Per Record or Per Collection mode.
3. Select Python Expression Editor in the Option Editor pane, and click the *Launch Python Editor* button.

Related Information

[Smart editor \[page 299\]](#)

14.2.2 Opening the Python Expression editor from the Match transform

Before you use the Python Expression editor, define your input and output fields for the Match transform in the transform editor.

1. In the Match transform editor *Options* tab, click the *Edit Options* button to open the Match Editor.

You can also select the Match transform in the data flow and choose ► *Tools* ► *Match Editor* ►.

2. Add a best record operation to your Match transform and select the appropriate option values.
3. To customize the Python code, make sure that you select *Yes* in the Custom column of the Best Record Action Fields table. Otherwise, the Python code is not editable.
4. Click the *View/Edit Python* button.

Related Information

[Smart editor \[page 299\]](#)

14.2.3 Writing Python code

In the Python Expression editor, create and edit Python code in the editor pane of the window. The Python expression that you create here depends on what you need to do with the Match or User-Defined transform.

The Python Expression editor includes keyword highlighting, auto-completion, auto-indentation, and code tool tips. As you type, the Python Expression editor highlights the correct Python syntax. It also auto-completes:

- The Python objects, functions, classes, and methods
- The SAP Data Services generated variables

Python API

The Python API tab lists the objects, functions, classes, and methods that are available for the specific transform and processing mode. When you select an item in the tab, information about it appears in the help area.

Input and output fields

The I/O Fields tab displays the input fields and output fields that have been mapped in the User-Defined transform. You can also add, delete, and edit the properties of user-defined input and output fields from this tab by right-clicking *Input Fields* or *Output Fields* and selecting *Insert*, *Delete*, or *Properties*.

The field variable name is the Mapped_Name option. For example, in a transform you may have an input field as follows:

- *Mapped_Name*: BEST_PRIMNAME1_IN

You could also have an output field as follows:

- *Mapped_Name*: BEST_PRIMNAME1_OUT

The Python Expression editor works in this way for both Per Record and Per Collection processing modes.

Custom options

For the User-Defined transform, you can create custom options that are used as variables within the transform (for example, a file path). You create the custom option in the User-Defined editor, and it is then displayed in the Variables tab of the Python Expression editor. Like substitution parameters, custom options are assigned \$\$ as a prefix and are enclosed in brackets (for example, [\$\$PATH]).

If you have both custom options and substitution parameters in your data flow, substitution parameters take precedence over custom options.

Related Information

[Smart editor \[page 299\]](#)

[Python examples \[page 1421\]](#)

14.2.4 Validating syntax

When you click the Validate button in the Python Expression editor, the syntax checker makes sure that the Python code has:

- All required colon (:) characters
- All string literal closing characters (either double quotes or single quotes)
- Correct indentation

Validating the Python syntax cannot prevent all runtime errors from occurring. Even if the code is syntactically correct, it might not execute correctly, in which case errors are generated during execution. The syntax checker cannot look for the incorrect usage of:

- Variable names
- Arguments to a function
- Method name on an object

14.2.5 Fixing syntax

If a syntax error is found, a message appears in the bottom section of the edit pane. The message points out the line and character number of the error.

To fix the syntax:

1. Double-click the error message. The Python Expression editor puts the focus on the specified line in the Python code.
2. After you fix an error, click the [Validate](#) button again. Messages are displayed one at a time; you may have additional syntax errors to fix.
3. Repeat steps 1 and 2 until all of the syntax is correct.

14.2.6 Find and Replace

Instead of browsing through lines of code, click the [Find](#) and [Replace](#) buttons to search for specific text and, if you want, replace it with other text. If the text is found, it is highlighted in the script.

Select the options [Match case](#) or [Match whole word only](#) to customize your search.

14.3 Built-in objects

Data Services includes internally-coded objects that you can use when writing expressions. Make sure to use the exact capitalization as you see it in this documentation. Python is case-sensitive.

The User-Defined transform supports two processing modes:

- Per Record
- Per Collection

Class	Object	Description	User-Defined Record Mode	User-Defined Collection Mode	Best Record (Match transform)
FLData Collection	Collection	Reference to each collection being processed.		Yes	
FLData Record	DST	Reference to destination record in a group posting operation.			Yes
	record	Reference to each record being processed by the data flow.	Yes		
	SRC	Reference to source record in a group posting operation.			Yes
FLData Manager	DataManager	Reference to the data manager for the data flow.		Yes	
FL Properties	Properties	Reference to a properties object.	Yes	Yes	Yes
FLPython String	RET	Value you want to post in a group posting operation.			Yes

14.4 Defined classes and methods

The following table lists every SAP Data Services-defined class and its supported methods in the User-Defined transform and the Best Record operation of the Match transform. The User-Defined transform supports two processing modes:

- Per Record
- Per Collection

Class	Method	User-Defined Record Mode	User-Defined Collection Mode	Best Record (Match transform)
FIData Collection	AddRecord		Yes	
	DeleteRecord		Yes	
	GetRecord		Yes	
	Size		Yes	
	Truncate		Yes	
FIData Manager	DeleteDataRecord		Yes	
	NewDataRecord		Yes	
FIData Record	GetField	Yes	Yes	Yes
	SetField	Yes	Yes	Yes
FIProperties	GetProperty	Yes	Yes	Yes
FIPython String	GetBuffer			Yes
	Set Buffer			Yes

14.5 FIDataCollection class

Use FIDataCollection class methods when you want to manipulate entire collections of data or records, and when adding new records to a collection that did not exist before. This can be helpful when matching in a real-time environment.

Related Information

[AddRecord \[page 1408\]](#)

[DeleteRecord \[page 1410\]](#)

[GetRecord \[page 1410\]](#)

[Size \[page 1411\]](#)

[Truncate \[page 1412\]](#)

14.5.1 AddRecord

≡ Syntax

```
AddRecord(<record>)
```

Description

Adds the record to the new collection.

i Note

For every `NewDataRecord()`, you can call `AddRecord()` only once.

i Note

After you call `AddRecord()`, do not call `DeleteDataRecord()`.

Parameters

This method has the following parameter.

Parameter	Description
record	Substitute the name of the record you want to add. This parameter is a variable that you must define.

Return value

None.

Example

In this example, the fields are defined for a new record and then the record is added to the collection with this `AddRecord()` method.

```
aDup = [1, u'brian', u'boyd', u'123 main st', u'83301'], \
[2, u'bryan', u'boyde', u'456 first st', u'83302'], \
[3, u'brina', u'boyle', u'789 last ave', u'83303']
for rec in aDup:
    newrecord = DataManager.NewDataRecord(1)
    newrecord.SetField(u'ID',unicode(rec[0]))
    newrecord.SetField(u'FIRST_NAME',unicode(rec[1]))
    newrecord.SetField(u'LAST_NAME',unicode(rec[2]))
    newrecord.SetField(u'ADDRESS',unicode(rec[3]))
    newrecord.SetField(u'POSTCODE1',unicode(rec[4]))
    Collection.AddRecord(newrecord)
```

14.5.2 DeleteRecord

Syntax

```
DeleteRecord(<record>)
```

Description

Removes the specified record from the collection.

Parameters

This method has the following parameter.

Parameter	Description
record	Substitute the name of the record you want to delete. This parameter is a variable that you must define.

Return value

None.

Example

In this example, a new record (newRec) is created and then deleted.

```
newRec = DataManager.NewDataRecord()  
Collection.DeleteRecord(newRec)
```

14.5.3 GetRecord

Syntax

```
GetRecord(<record>, <index>)
```

Description

Retrieves the value of a record in a collection in the specified index position.

Parameters

This method has the following parameters.

Parameter	Description
record	Substitute the name of the record object. This parameter is a variable that you must define; for example, <code>record = DataManager.NewDataRecord()</code>
index	Substitute the numerical index value of the record in the collection.

Return value

Returns the value from the record at the specified position.

Example

In the following example, a new record (newRec) is created, the value of the record in position one is retrieved, and the record is deleted from the collection.

```
newRec = DataManager.NewDataRecord()  
Collection.GetRecord(newRec, 1)  
Collection.DeleteRecord(newRec)
```

14.5.4 Size

≡ Syntax

Size()

Description

Counts the number of records in the collection.

Parameters

None.

Return value

Returns an integer that refers to the number of records in the collection.

Example

In this example, you're retrieving the number of records in the collection.

```
collectionSize = Collection.Size()
```

14.5.5 Truncate

≡ Syntax

```
Truncate()
```

Description

Removes all records from a collection, but does not delete the collection.

Parameters

None.

Return value

None.

Example

Use this method to quickly delete all the records from a collection, rather than one by one.

```
Collection.Truncate()
```

14.6 FIDataManager class

Use FIDataManager class methods when you want to create new records.

Related Information

[DeleteDataRecord \[page 1413\]](#)

[NewDataRecord \[page 1414\]](#)

14.6.1 DeleteDataRecord

≡ Syntax

```
DeleteDataRecord(<record>)
```

Description

Deletes the memory of a record object allocated using NewDataRecord().

i Note

Do not call `DeleteDataRecord()` after calling `AddRecord()`.

Parameters

This method has the following parameter.

Parameter	Description
record	Substitute the name of the record object you want to delete.

Return value

None.

Example

In this example, a new record object (`newRec`) is created. Then, using this method, the memory allocated to the data collection is deleted. You must use this method when you use the `NewDataRecord()` method, otherwise the Python expression may have a memory leak.

```
newRec = DataManager.NewDataRecord()  
DataManager.DeleteDataRecord(newRec)
```

14.6.2 NewDataRecord

Syntax

```
NewDataRecord()
```

Description

Creates a new record object. Do not use this method in a loop, otherwise the Python expression may experience a memory leak. Depending on the expression, you'll probably want to place this method at the beginning of the expression.

Parameters

This method has none, but see example for an exception.

If you call the record with a parameter of 1, then the new record gets its own memory.

Return value

Returns a new object of type `FIDataRecord`.

Examples

In the following example, a new record (`newRecord`) is created and populated in the original collection.

```
newRecord = DataManager.NewDataRecord()
#gets the number of records
numRecords = Collection.Size()
#iterate over the collection
for recordNum in range (1, numRecords + 1)
    #get a record
    Collection.GetRecord(newRecord, recordNum)
    #set a field on the record
    newRecord.SetField(u'NAME', u'test')

DataManager.DeleteDataManager (newRecord)
```

The following example is a little different from the previous one. In this example, records are read from a database and then are added to the original collection. Because of this difference, the `NewDataRecord()` method then needs a numeric parameter of 1.

```
newRecord = DataManager.NewDataRecord(1)
#get the records from the database (excluded from example)
#populate the record
newRecord.SetField(u'NAME', u'test')
#add record to the collection
Collection.AddRecord(newRecord)
del newRecord
```

⚠ Caution

Make sure that you clean up memory references.

Related Information

[About Python \[page 1400\]](#)

14.7 FIDataRecord class

Use Data Services-defined FIDataRecord class methods to manipulate existing individual records.

Related Information

[GetField \[page 1416\]](#)

[SetField \[page 1417\]](#)

14.7.1 GetField

Syntax

```
fieldVal = GetField(u'<fieldName>')
```

Description

Retrieves the contents of the specified input field. This method can be used with defined input fields only.

Parameters

This method has the following parameter:

Parameter	Description
fieldName	<p>In the Python Expression editor, use one of the input field variables.</p> <p>If you use this method with the Best Record operation of the Match transform, this parameter should be replaced with the Mapped Name you want to retrieve.</p>

Return value

Returns a new string with the contents of the specified field.

Example

```
if newRecord.GetField(u'POSTCODE1') == u'54601'...
```

⚠ Caution

Make sure to use a "u" to indicate Unicode every time you use a Unicode string to look up field names. If you do not, an error or crash may occur.

14.7.2 SetField

≡ Syntax

```
SetField(u'<fieldName>', u'value')
```

Description

Stores a value in the specified field.

Parameters

This method has the following parameters:

Parameter	Description
fieldName	In the Python Expression editor, use one of the input or output field variables, which uses the Mapped Name.
value	Specifies the value you want to store in the field.

Return value

None.

Example

For example, you could store "Current Resident" in the field named NAME_SUBSTITUTION.

```
newRecord.SetField(u'NAME_SUBSTITUTION', u'Current Resident')
```

⚠ Caution

Make sure to use a "u" to indicate Unicode every time you use a Unicode string to look up field names. If you do not, an error or crash may occur.

14.8 FIProperties class

Use the FIProperties class to gain access to various properties of the system that the Python expression is running in. The class can access the following run-time parameters in the Data Services environment variables:

Run-time parameter	Description
APPLICATION_PATH	The directory that contains the application executable.
APPLICATION_VERSION	The version of the framework.
DATAFLOW_NAME	The name of a data flow.
JOB_ID	The run ID of a job.
TRANSFORM_GUID	The globally unique identifier, or GUID, of a transform
TRANSFORM_NAME	The display name of a transform.
REPOSITORY_VERSION	The version of a repository.

Related Information

[GetProperty \[page 1418\]](#)

14.8.1 GetProperty

≡ Syntax

```
var1 = GetProperty(<PropertyName>)
```

Description

Returns the value of a given property specified as an input parameter.

Parameters

This method has the following parameter:

Parameter	Description
PropertyName	Specifies the environment variable that you want to retrieve.

Return value

Returns the value of the specified property.

Example

The following example shows how to retrieve a value for the JOB_ID parameter.

```
#Retrieve the Property Value for JOB_ID
propValue = Properties.GetProperty(u'JOB_ID')
#Set the Job Id value into JOB_ID_OUT field
record.SetField(u'JOB_ID_OUT',unicode(propValue))
del propValue
```

14.9 FIPythonString class

Use the FIPythonString methods to customize your data processing. With these methods, you can create a Best Record operation in the Match transform.

Related Information

[GetBuffer \[page 1420\]](#)

[SetBuffer \[page 1420\]](#)

14.9.1 GetBuffer

≡ Syntax

```
GetBuffer()
```

Description

Returns the specified string.

Parameters

None.

Return value

The Unicode character string.

Example

In the following example, [getstr](#) would hold the value of the Unicode character string.

```
getstr = STR.GetBuffer()
```

14.9.2 SetBuffer

≡ Syntax

```
SetBuffer(u'<stringValue>')
```

Description

Sets character buffer to the object.

Parameters

This method has the following parameter:

Parameter	Description
stringValue	Specifies the string that you want to use here.

Return value

None.

Example

This example shows how to post data from a “master” record to its “subordinates” with the Best Record operation of the Match transform. The data is input with the field input.code.

Best Record strategy:

```
# store master and subordinate values
SOURCE = SRC.GetField(u'input.code')
DESTINATION = DST.GetField(u'input.code')
# if the master is not empty and the subordinate is
if len(SOURCE.strip()) != 0 and len(DESTINATION.strip()) == 0:
    RET.SetBuffer(u'T')
else:
    RET.SetBuffer(u'F')
# delete temporary variables
del SOURCE
del DESTINATION
```

Best Record Action:

```
# store master
SOURCE = SRC.GetField(u'input.code')
# return master
RET.SetBuffer(SOURCE)
# delete temporary variables
del SOURCE
```

14.10 Python examples

The following examples, grouped by the type of action they perform on data, are intended to help you get started writing expressions in Python. You may need to significantly change some of these examples to fit the type of data and names of fields you are using.

Keep in mind that many of these tasks could be also performed with a Query transform.

14.10.1 Formatting data

The following examples of Python code in the User-Defined transform can be used to format data.

Example use	Sample Python code
Data is input without data-source identification. In the User-Defined transform, append a field mapped to the source and populate it with TRC for all records.	<pre>record.SetField(u'SOURCE',u'TRC')</pre>
Data is input with a name field mapped to name. In the User-Defined transform, upper case the name and put it in a new name field that is mapped to uppername.	<pre>name = record.GetField(u'name') uppername = name.upper() record.SetField(u'uppername',unicode(uppername)) del name del uppername</pre>
Data is input with an account type indicator field, mapped to account_type, that contains B or b for business accounts, and I or I for individual accounts. In the User-Defined transform, append two fields mapped to name and firm. If records contain B or b, output the contents of the field mapped to customer_name to the new firm field. If records do not contain B or b, output the contents to the new name field.	<pre>account_type = record.GetField(u'acct_type') customer_name = record.GetField(u'cust') if account_type.strip().upper() == u'B': record.SetField(u'firm',unicode(customer_name)) else: record.SetField(u'name',unicode(customer_name)) del account_type del customer_name</pre>
Firm data is input in a field mapped to firm. In the User-Defined transform, populate a two-character field mapped to firm_length that contains the number of characters in the firm name (padded with zeros).	<pre>field = record.GetField(u'firm') firm_length = field.strip().zfill(2) record.SetField(u'firm_length',unicode(firm_length)) del field del firm_length</pre>
Data is input with some records not having a name in the field mapped to name. In the User-Defined transform, complete empty names with Valued Customer, preserving the input name in records that have them. Overwrite the data in the same field.	<pre>name_in = record.GetField(u'name') if len(name_in.strip()) == 0: record.SetField(u'name',u'Valued Customer') del name_in</pre>
In this example, name_in is the mapped name for the input name field. In the output field section, name_out is the mapped name for the same field.	

Example use

Sample Python code

Data is input with fields mapped to ZIP Code and street. In the User-Defined transform, append a field mapped to breakgroupid, and populate it with the first three characters of the ZIP Code and the first three characters of the street.

```
zip = record.GetField(u'zip')
street = record.GetField(u'street')
breakgroupid = zip[0:3] + street[0:3]
record.SetField(u'breakgroupid', unicode(breakgroupid))
del zip
del street
del breakgroupid
```

Data is input with a field mapped to groupnumber. In the User-Defined transform, append a field mapped to groupnumberzeropad, and populate it with the group number, padded with zeros to 10 characters in length.

```
groupnumber = record.GetField(u'groupnumber')
groupnumberzeropad =
groupnumber.strip().zfill(10)
record.SetField(u'groupnumberzeropad', unicode(groupnumberzeropad))
del groupnumber
del groupnumberzeropad
```

In the User-Defined transform, append a field mapped to recordnum, and populate it with the record number.

```
dct = locals()
if dct.has_key('COUNTER'):
    dct['COUNTER'] = dct['COUNTER'] + 1
else:
    dct['COUNTER'] = 1
record.SetField(u'recordnum', unicode(dct['COUNTER']))
```

In the User-Defined transform, append a field mapped to recordnum, and populate it with the record number, zero padded to 10 characters in length.

```
dct = locals()
if dct.has_key('COUNTER'):
    dct['COUNTER'] = dct['COUNTER'] + 1
else:
    dct['COUNTER'] = 1
recordnum = str(dct['COUNTER']).zfill(10)
record.SetField(u'recordnum', unicode(recordnum))
del recordnum
```

Data is input in user_group and user_code fields. In the User-Defined transform, if the contents of user_code is A, B, C, D, E, F, G, or H, output UserGroupA in the user_group field. If user_code contains I, J, K, L, M, N, O, or P, output UserGroupB in the user_group field. If user_code contains any other value, preserve the input value in the user_group field.

```
user_code = record.GetField(u'user_code')
uga = 'A,B,C,D,E,F,G,H'
ugb = 'I,J,K,L,M,N,O,P'
if uga.find(user_code.strip().upper()) > -1:
    record.SetField(u'user_group', u'UserGroupA')
elif ugb.find(user_code.strip().upper()) > -1:
    record.SetField(u'user_group', u'UserGroupB')
del user_code
del uga
del ugb
```

14.10.2 Splitting data

The following examples can be used in the User-Defined transform to split your data in a specific way without changing how it is routed.

Example use

Sample Python code

Data is input in an input account field (mapped to account) with contents of name, a slash, and firm, for example "John Smith / SAP". In the User-Defined transform, append two new fields mapped to name and firm, where the contents before the slash are placed in name and the contents after the slash are placed in firm.

In this example, the syntax str specifies the type of split and the syntax 3-part specifies how to split.

```
from flscansplit import ScanSplit
account = record.GetField(u'account')
name = ScanSplit(account, u'str', u'3-part', ['/'])[0]
firm = ScanSplit(account, u'str', u'3-part', ['/'])[2]
record.SetField(u'name',unicode(name))
record.SetField(u'firm',unicode(firm))
del account
del name
del firm
```

Data is input in an input.account field (mapped to account) with contents of a person's name followed by a financial suffix, for example "John Smith JTWROS". In the User-Defined transform, append two new fields mapped to account_name and account_type, where the name and type are split.

```
from flscansplit import ScanSplit
account = record.GetField(u'account')
type = [u'JT/WROS', u'JT WROS', u'JTWROS', u'JT/TEN', u'JT TEN', u'JT TEN', u'JT TIC', u'JT TIC', u'JT TIC', u'JT TIC', u'TEN COM', u'TEN/COM', u'TENCOM']
account_name = ScanSplit(account, u'str', u'before',type)[0]
account_type = ScanSplit(account, u'str', u'before',type)[1]
record.SetField(u'account_name',unicode(account_name))
record.SetField(u'account_type',unicode(account_type))
del account
del type
del account_name
del account_type
```

Another option for scan values is to create an external file in a text editor and saved locally, with extension "py". Import the file prior to the method. Then in the ScanSplit method, use the variable in the file in place of the actual scan values.

For example, to accomplish the same account_name and account_type fields specified in the second example, you may create a file called suffixes.py that has the following contents:

```
type = [u'JT/WROS', u'JT WROS', u'JTWROS', u'JT/TEN', u'JT TEN', u'JT TEN', u'JT TIC', u'JT TIC', u'JT TIC', u'JT TIC', u'TEN COM', u'TEN/COM', u'TENCOM']
```

Then, complete the following expression in the User-Defined transform.

```
from flscansplit import ScanSplit
from suffixes import *
account = record.GetField(u'account')
account_name = ScanSplit(account, u'str', u'before',type)[0]
account_type = ScanSplit(account, u'str', u'before',type)[1]
record.SetField(u'account_name',unicode(account_name))
```

```
record.SetField(u'account_type', unicode(account_type))
del account
del account_name
del account_type
```

14.10.3 Best Record

The following example can be used in the Best Record operation in the Match transform. This example shows the use of Unicode.

Example use

Data is input with a field `gen.phone` that is populated in some records of a match group and empty in others. Perform the best record action of taking phone data from a populated record and placing it into an empty record.

In this example, the Best Record strategy returns a `True` when the source is populated and the destination is empty (or else, it returns a `False`). At the end, the `GetField` method places the source data into the destination field, provided the Best Record strategy returns a `True`.

Sample Python code

Best Record strategy:

```
Source = SRC.GetField(u'gen.phone')
Destination = DST.GetField(u'gen.phone')
if len(Source.strip()) > 0 and
len(Destination.strip()) == 0:
    RET.SetBuffer(u'T')
else:
    RET.SetBuffer(u'F')
del Source
del Destination
```

Best Record Action:

```
RET.SetBuffer(SRC.GetField(u'gen.phone'))
```

14.10.4 Assigning source attributes

The following example can be used in the User-Defined transform to assign a source to records in a collection.

Example use

Data is input with a field mapped to SOURCE_IN. When the source is CRM or LEADS, assign source attributes. When the source is DoNotMarket, assign list attributes.

Sample Python code

```
SOURCE_IN = SRC.GetField(u'SOURCE_IN')
if SOURCE_IN.strip() == u'CRM':
    SOURCE_TYPE_OUT = u'N'
    DRIVER_ORDER_OUT = u'020'
    BEST_RECORD_PRIORITY_OUT = u'010'
    INCLUDE_IN_SOURCE_COUNT_OUT = u'Y'
    APPLY_BLANK_PENALTY_OUT = u'Y'
    PERFORM_DATA_SALVAGE_OUT = u'N'
    PROTECT_UNIQUE_ID_OUT = u'Y'
elif SOURCE_IN.strip() == u'Leads':
    SOURCE_TYPE_OUT = u'N'
    DRIVER_ORDER_OUT = u'010'
    BEST_RECORD_PRIORITY_OUT = u'020'
    INCLUDE_IN_SOURCE_COUNT_OUT = u'Y'
    APPLY_BLANK_PENALTY_OUT = u'Y'
    PERFORM_DATA_SALVAGE_OUT = u'N'
    PROTECT_UNIQUE_ID_OUT = u'N'
elif SOURCE_IN.strip() == u'DoNotMarket':
    SOURCE_TYPE_OUT = u'S'
    DRIVER_ORDER_OUT = u'000'
    BEST_RECORD_PRIORITY_OUT = u'000'
    INCLUDE_IN_SOURCE_COUNT_OUT = u'Y'
    APPLY_BLANK_PENALTY_OUT = u'N'
    PERFORM_DATA_SALVAGE_OUT = u'N'
    PROTECT_UNIQUE_ID_OUT = u'N'
record.SetField(u'SOURCE_TYPE_OUT', unicode(SOURCE_TYPE_OUT))
record.SetField(u'DRIVER_ORDER_OUT', unicode(DRIVER_ORDER_OUT))
record.SetField(u'BEST_RECORD_PRIORITY_OUT', unicode(BEST_RECORD_PRIORITY_OUT))
record.SetField(u'INCLUDE_IN_SOURCE_COUNT_OUT', unicode(INCLUDE_IN_SOURCE_COUNT_OUT))
record.SetField(u'APPLY_BLANK_PENALTY_OUT', unicode(APPLY_BLANK_PENALTY_OUT))
record.SetField(u'PERFORM_DATA_SALVAGE_OUT', unicode(PERFORM_DATA_SALVAGE_OUT))
record.SetField(u'PROTECT_UNIQUE_ID_OUT', unicode(PROTECT_UNIQUE_ID_OUT))
del SOURCE_TYPE_OUT
del DRIVER_ORDER_OUT
del BEST_RECORD_PRIORITY_OUT
del INCLUDE_IN_SOURCE_COUNT_OUT
del APPLY_BLANK_PENALTY_OUT
del PERFORM_DATA_SALVAGE_OUT
del PROTECT_UNIQUE_ID_OUT
```

15 Hadoop overview

Use SAP Data Services to connect to Apache Hadoop frameworks, including Hadoop Distributive File Systems (HDFS) and Hive sources and targets.

Data Services supports Hadoop on both the Linux and Windows platform. For Windows support, Data Services uses Hortonworks Data Platform (HDP) only. HDP allows data from many sources and formats. See the latest *Product Availability Matrix (PAM)* for the supported versions of HDP.

For information about deploying Data Services on a Hadoop MapR cluster machine, see SAP Note [2404486](#).

i Note

To find out what Hadoop features are supported in your current version of Data Services, see [Hadoop in Data Services \[page 1429\]](#).

The following table describes the relevant components of Hadoop:

Component	Description
Hadoop Distributed File System (HDFS)	A distributive file system that stores data on nodes, providing high aggregate bandwidth across the cluster.
Hive	A data warehouse infrastructure that allows SQL like ad hoc querying of data (in any format) stored in Hadoop.
Pig	A high-level data flow language and execution framework for parallel computation that is built on top of Hadoop. Data Services uses Pig scripts to read from and write to HDFS, including join and push-down operations.
Map/Reduce	A computational paradigm where the application is divided into many small fragments of work, each of which may be executed or re-executed on any node in the cluster. Data Services uses map/reduce to do text data processing.

The following table describes all of the Hadoop-related objects in Data Services that you use to work with your Hadoop data.

Hadoop objects and tools

Object	Description
Hive adapter	Enables Data Services to connect to a Hive server so that you can work with data from Hadoop. For complete information about using Data Services adapters, see the <i>Supplement for Adapters</i> .

Object	Description
Hive datastore	<p>Enables Data Services to access data from your Hive data warehouse to use the data as a source or a target in Data Services processing.</p> <p>There are two types of Hive datastores:</p> <ul style="list-style-type: none"> Hive adapter datastore: Use with the Hive adapter. <div> i Note Install Data Services on the machine within the Hadoop cluster to use the Hive adapter datastore. </div> <ul style="list-style-type: none"> Hive database datastore. <div> i Note Install Data Services on any machine. </div> <p>To access a remote Hive server, configure the Hive database datastore with a supported Hive ODBC driver and a data source name (DSN).</p> <p>For complete information about adapters and creating an adapter datastore, see the <i>Supplement for Adapters</i> and the <i>Reference Guide</i>.</p>
HDFS file format	<p>Contains a description for your HDFS file system structure.</p> <div> i Note To use as a source or target, install Data Services within the Hadoop cluster. Use an HDFS file location to connect to HDFS when Data Services is installed outside of the Hadoop cluster. </div>
HDFS file location	<p>Contains connection information to your HDFS.</p> <p>To use the HDFS file location as a source or target in a data flow, use a file format that is not an HDFS file format. For example, use a flat file format.</p> <div> i Note Install Data Services on any machine. </div>

Object	Description
Hive template table	<p>Use a Hive template table as a target using one of the following two methods:</p> <ul style="list-style-type: none"> Method 1: Use a Hive datastore template table from the Datastore tab in the Designer object library. Method 2: Use a template table from the Designer tool palette. <p>After you have used a Hive template table in an executed data flow, you can use the target Hive template as a source in a data flow.</p>

Related Information

[Using an ODBC driver to connect to a remote Hive Server \[page 82\]](#)

[Template table \[page 274\]](#)

15.1 Hadoop in Data Services

SAP Data Services has added support for Hadoop in stages, with features added in specific Data Services versions.

If you have not upgraded to the most recent release of Data Services, use the following table to determine how your version of Data Services supports Hadoop.

Hadoop features in chronological order

Data Services version	Hadoop support	More information
4.2 SP1	<p>Connect to Hadoop Hive and HDFS, Linux only:</p> <ul style="list-style-type: none"> HDFS file format Hive adapter Hive adapter datastore 	For more information about Data Services adapters, see the <i>Supplement for Adapters</i>

Data Services version	Hadoop support	More information
4.2 SP2	<p>For the Hive adapter datastore, support for:</p> <ul style="list-style-type: none"> • Hive Server version 2 • Hive Server sub version 0.11 and later <p>Users must migrate to new Hive Server versions.</p>	For current version information, consult the Product Availability Matrix (PAM).
4.2 SP3	Preview Hive table data	<i>Supplement for Hadoop</i>
4.2 SP4	Preview HDFS data	<i>Supplement for Hadoop</i>
4.2 SP5	Hive datastore support for SQL functions and SQL transform.	<i>Supplement for Hadoop</i>
4.2 SP6	<ul style="list-style-type: none"> • Hive template table • SASL-QoP with Kerberos on Hive (Simple Authentication and Security Layer - Quality of Protection) • Windows support for Hadoop in Data Services • Hive support for Varchar and Char data types 	<i>Supplement for Hadoop</i> <i>Reference Guide</i>
4.2 SP7	<ul style="list-style-type: none"> • Hive on Spark • Support for Hive Beeline CLI to replace Hive CLI • Deprecated support for Hive CLI 	<i>Supplement for Hadoop</i>
4.2 SP8	<ul style="list-style-type: none"> • Support Edge nodes running on Hadoop Hive clusters, Linux only • Hive template tables that use Parquet, AVRO, and ORC table formats. 	<i>Supplement for Hadoop</i>
4.2 SP9	Connect to Hadoop on SAP Cloud Platform Big Data Services	<i>Supplement for SAP Cloud Platform Big Data Services</i>

Data Services version	Hadoop support	More information
4.2 SP10	<p>HDFS file location object:</p> <ul style="list-style-type: none"> Supports reading and loading to HDFS system with or without Kerberos authentication. Data Services installation is not restricted to be inside the Hadoop cluster. Install Data Services on any machine for HDFS and Hive reading and loading. <p>Hive datastore enhancements:</p> <ul style="list-style-type: none"> Use supported Hive ODBC drivers to remotely connect to Hive server. Data Services installation is not restricted to be inside the Hadoop cluster. Support for Kerberos. Support for bulk loading to Hive. 	<p><i>Supplement for Adapters</i></p> <p><i>Supplement for Hadoop</i></p> <p><i>Reference Guide</i></p>

15.2 Hadoop sources and targets

Use SAP Data Services objects that you configure for Hive or Hadoop Distributive File System (HDFS) as sources and targets in data flows.

To access data from Hive, use objects that are designed for Hive. For example, use the Hive adapter datastore for jobs that use data from your Hive storage. When you want data from your HDFS, use an HDFS file format or HDFS file location object.

Use other Data Services objects along with Hadoop objects in data flows based on your objectives.

❖ Example

- Configure a data source name (DSN) using a supported Hive ODBC driver to create a Hive datastore that accesses a remote Hive server.
- Configure a flat file with an HDFS file location object and use as a source or target in a data flow.
- Use an HDFS file location object and a script to access data from a remote source with the `copy_from_remote_system` function.
- Use an HDFS file location object and a script to upload data from your local server to a remote server using the `copy_to_remote_system` function.
- Use bulk loading to upload data to Hive or HDFS. Works with a flat file, Hive template table, or a table within the Hive datastore as a target in a data flow.

Related Information

[Connect to HDFS \[page 1437\]](#)

[Connect to Hive \[page 1438\]](#)

[Using an ODBC driver to connect to a remote Hive Server \[page 82\]](#)

[Configuring bulk loading for Hive \[page 84\]](#)

15.3 Prerequisites to Data Services configuration

Before configuring SAP Data Services for connecting to Hadoop, verify that your system configuration is correct.

Ensure that your Data Services system configuration meets the following prerequisites.

For Linux and Windows platforms:

- Make sure the machine where the Data Services Job Server is installed is configured to work with Hadoop.
- Make sure the machine where the Data Services Job Server is installed has the Pig client installed.
- If you are using Hive, verify that the Hive client is installed. To verify this, log on to the node and issue Pig and Hive commands that invoke the respective interfaces.
- Install the Data Services Job Server on one of the Hadoop cluster machines, which can be either an Edge (Linux only) or a Data node.

i Note

To install Data Services to any machine, including the machine in the cluster, use a supported ODBC driver in the Hive adapter datastore, or set up jobs using the HDFS file location object. Create the HDFS file location object using either the WebHDFS or the HTTPFS connection protocols.

- If you are using text data processing, ensure that you have copied the necessary text data processing components to the HDFS, which enables MapReduce functionality.

For Linux platforms:

- Ensure that the environment is set up correctly for interaction with Hadoop.
- Start the Job Server from an environment that has sourced the Hadoop environment script. For example:

```
source <LINK_DIR>/hadoop/bin/hadoop_env_setup.sh -e
```

Related Information

[HDFS file location object options \[page 187\]](#)

[Hive datastores \[page 78\]](#)

15.4 Verify Linux setup with common commands

Use common commands to verify that your SAP Data Services system on Windows is correctly configured for Hadoop.

When you use the commands in this topic, your output may be different than what we show. That is okay as long as your commands do not result in errors.

Setting up the environment

To set up the Data Services environment for Hadoop, use the following command:

```
$ cd <DS Install Directory>/bin
$ source ./al_env.sh
$ cd ../hadoop/bin
$ source ./hadoop_env_setup.sh -e
```

Checking components

Ensure that Hadoop, Pig, and Hive are installed and correctly configured on the machine where Data Services Job Server for Hadoop resides.

To make sure that Hadoop, Pig, and Hive are set up correctly, use the following command:

```
$ hadoop fs -ls /
```

For Hadoop, you should see output similar to the following:

```
$ hadoop fs -ls /
Found 2 items
drwxr-xr-x - hadoop supergroup          0 2013-03-21 11:47 /tmp
drwxr-xr-x - hadoop supergroup          0 2013-03-14 02:50 /user
```

For Pig, you should see output similar to the following:

```
$ pig
INFO org.apache.pig.Main - Logging error messages to: /hadoop/
pig_1363897065467.log
INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine -
Connecting to hadoop file system at: hdfs://machine:9000
INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine -
Connecting to map-reduce job tracker at: machine:9001
grunt> fs -ls /
Found 2 items
drwxr-xr-x - hadoop supergroup          0 2013-03-21 11:47 /tmp
drwxr-xr-x - hadoop supergroup          0 2013-03-14 02:50 /user
grunt> quit
```

For Hive, you should see output similar to the following:

```
$ hive
```

```
Hive history file=/tmp/hadoop/hive_job_log_hadoop_201303211318_504071234.txt
hive> show databases;
OK
default
Time taken: 1.312 seconds
hive> quit;
```

Set up or restart the Job Server

If all commands pass, use the following command from within the same shell to set up or restart the Job Server.

```
<LINK_DIR>/bin/svrcfg
```

This command provides the Job Server with the proper environment from which it starts engines that call Hadoop, Pig, and Hive.

15.4.1 Configuring Hadoop for text data processing

SAP Data Services supports text data processing in the Hadoop framework using a MapReduce form of the Entity Extraction transform.

To use text data processing in Hadoop, run the following Hadoop environment script.

```
<LINK_DIR>/hadoop/bin/hadoop_env_setup.sh -c
```

The script copies the language modules and other dependent libraries to the Hadoop file system so that MapReduce can distribute them during the MapReduce job setup. You only have to do this file-copying operation once after an installation or update, or when you want to use custom dictionaries or rule files.

If you use the Entity Extraction transform with custom dictionaries or rule files, copy these files to the Hadoop file system for distribution. To do so, first copy the files into the languages directory of the Data Services installation, then rerun the Hadoop environment script. For example:

```
cp /myhome/myDictionary.nc <LINK_DIR>/TextAnalysis/languages
```

```
<LINK_DIR>/hadoop/bin/hadoop_env_setup.sh -c
```

After you complete the Hadoop environment set up, you can have the Entity Extraction transform operations pushed down and handled by the Hadoop system by connecting it to a single HDFS Unstructured Text source.

Optimizing text data processing for use in the Hadoop framework

With text data processing in the Hadoop framework, use the Hadoop configuration setting `mapred.max.split.size` to control the following:

- Amount of data a mapper can handle
- Number of mappers used by a job

Set the value for `mapred.max.split.size` in the Hadoop configuration file. The Hadoop configuration file is located at `$HADOOP_HOME/conf/core-site.xml`. The Hadoop configuration file could be located in an alternate location, depending on the type of Hadoop you use.

By default, the value for `mapred.max.split.size` is 0. A setting of 0 means that there is no limit to the amount of data the mapper can handle, and text data processing runs with only one mapper. Change the default configuration value to the amount of data each mapper can handle.

❖ Example

For example, a Hadoop cluster contains twenty machines. Each machine is set to run a maximum of ten mappers. $20 \times 10 = 200$ mappers available in the cluster.

Your input data averages 200 GB. To have the text data processing job consume 100 percent of the available mappers, set `mapred.max.split.size` to 1073741824 (1 GB). Calculation: $200 \text{ GB} \div 200 \text{ mappers} = 1 \text{ GB per mapper}$.

```
<property>
  <name>mapred.max.split.size</name>
  <value>1073741824</value>
</property>
```

To have the text data processing job consume 50 percent of the available mappers, set `mapred.max.split.size` to 2147483648 (2 GB). Calculation: $200 \text{ GB} \div 100 \text{ mappers} = 2 \text{ GB per mapper}$.

Related Information

[Entity Extraction transform \[page 1014\]](#)

15.5 Hadoop support for the Windows platform

SAP Data Services supports Hadoop on the Windows platform using Hortonworks.

Use the supported version of Hortonworks HDP only. See the *Product Availability Matrix (PAM)* for the most recent supported version number.

When you use Hadoop on the Windows platform, use Data Services to do the following tasks:

- Use Hive tables as a source or target in your data flows.
- Use HDFS files as a source or target in your data flows using Pig script or the HDFS library API.
- Use HDFS file location object as a source or target in your data flows. Pig Script or HDFS library API is not required.
- Stage non-Hive data in a data flow using the `Data_Transfer` transform.

- Preview data for HDFS files and Hive tables.

Requirements

Make sure that you set up your system as follows:

- Install the Data Services Job Server in one of the nodes of the Hadoop cluster.

i Note

Data Services can be installed on any machine when you use an HDFS file location object with WebHDFS or HTTPFS connection protocols, or when you use a Hive database datastore that is configured with DSN and a supported Hive ODBC driver.

- Set the system environment variables, such as PATH and CLASSPATH, so that the Job Server can run as a service.
- Set the HDFS file system permission requirements for using HDFS or Hive.

Related Information

[Connect to HDFS \[page 1437\]](#)

[Previewing HDFS file data \[page 1438\]](#)

[HDFS file location object options \[page 187\]](#)

[Hive datastores \[page 78\]](#)

15.5.1 Setting up HDFS and Hive on Windows

Set system environment variables and use command prompts to configure HDFS and Hive for Windows.

Install the SAP Data Services Job Server component.

Perform the following steps to set up HDFS and Hive on Windows:

1. Set the following system environment variable:

```
HDFS_LIB_DIR = /sap/dataservices/hadoop/tdp
```

2. Add the following to the PATH: `<LINK_DIR>\ext\jre\bin\server`.
3. Run the following command:

```
hadoop classpath --jar c:\temp\hdpclasspath.jar
```

4. Update `CLASSPATH=%CLASSPATH%; c:\temp\hdpclasspath.jar`.

CLASSPATH generates the Hadoop and Classpath .jar files.

5. Set the location of the Hadoop and Classpath .jar files.
6. When the Hadoop CLASSPATH command completes successfully, check the content of the .jar file for the Manifest file.
7. Check that the `hdfs.dll` has symbols exported.

If the symbols from the `hdfs.dll` are not exported, install the fix from Hortonworks for the export of symbols. If the symbols from the `.dll` are still not exported, use the `.dll` from Hortonworks 2.3.

8. Required only if you use Text Data Processing transforms in jobs, and only once per Data Services install: Run the following command from `<LINK_DIR>\bin`:

```
Hadoop_env_setup.bat
```

The `.bat` file copies the Text Analysis Language file to the HDFS cache directory.

9. Ensure that the Hadoop or Hive `.jar` files are installed. The Data Services Hive adapter uses the following `.jar` files:

- `commons-httpclient-3.0.1.jar`
- `commons-logging-1.1.3.jar`
- `hadoop-common-2.6.0.2.2.6.0-2800.jar`
- `hive-exec-0.14.0.2.2.6.0-2800.jar`
- `hive-jdbc-0.14.0.2.2.6.0-2800-standalone.jar`
- `hive-jdbc-0.14.0.2.2.6.0-2800.jar`
- `hive-metastore-0.14.0.2.2.6.0-2800.jar`
- `hive-service-0.14.0.2.2.6.0-2800.jar`
- `httpclient-4.2.5.jar`
- `httpcore-4.2.5.jar`
- `libfb303-0.9.0.jar`
- `log4j-1.2.16.jar`
- `slf4j-api-1.7.5.jar`
- `slf4j-log4j12-1.7.5.jar`

10. Run the following commands to set up the permissions on the HDFS file system:

```
hdfs dfs -chmod -R 777 /mapred
hdfs dfs -mkdir /tmp
hdfs dfs -chmod -R 777 /tmp
hdfs dfs -mkdir /tmp/hive/
hdfs dfs -chmod -R 777 /tmp/hive
hdfs dfs -mkdir -p /sap/dataservices/hadoop/tdp
hdfs dfs -mkdir -p /user/hive
hdfs dfs -mkdir -p /hive/warehouse
hdfs dfs -chown hadoop:hadoop /user/hive
hdfs dfs -chmod -R 755 /user/hive
hdfs dfs -chmod -R 777 /hive/warehouse
```

15.6 Connect to HDFS

Connect to your Hadoop Distributed File System (HDFS) data using an HDFS file format or an HDFS file location.

An HDFS file format and an HDFS file location contain your HDFS connection information, including account name, password, security protocol, and so on. Data Services uses this information to access HDFS data during Data Services processing.

Decide which object to use based on the location of your Data Services installation:

- Use an HDFS file format when Data Services is installed within the Hadoop cluster.
- Use an HDFS file location when Data Services is installed anywhere, including within the Hadoop cluster.

If your Hadoop system is managed in SAP Cloud Platform Big Data Service (formerly Altiscale), your connection setup uses information from your Big Data Service account. For complete instructions to connect to your Big Data Service account, see the *Supplement for SAP Cloud Platform Big Data Service*.

Related Information

[HDFS file location object options \[page 187\]](#)

15.6.1 Previewing HDFS file data

Preview HDFS file data for delimited and fixed width file types.

To preview the first 20 or so rows of an HDFS file:

1. Right-click an HDFS file name in the *Format* tab of the Local Object Library
2. Click *Edit*.

The *File Format Editor* opens. You can only view the data. Sorting and filtering are not available when you view sample data in this manner.

Use one of the following methods to access HDFS file data so that you can view, sort, and filter the data:

- Right-click on HDFS source or target object in a data flow and click *View Data*.
- Click the magnifying glass icon located in the lower right corner of the HDFS source or target objects in the data flow.
- Right-click an HDFS file in the *Format* tab of the Local Object Library, click *Properties*, and then open the *View Data* tab.

Note

By default, the maximum number of rows displayed for data preview and filtering is 1000, but you can adjust the number lower or higher, up to a maximum of 5000. To change the maximum number of rows to display:

1. Select **Tools > Options > Designer > General**.
2. Set the *View data sampling size (rows)* to the desired number of rows.

15.7 Connect to Hive

Use the Hive adapter and the Hive datastore to connect to Hive.

When you create the Hive adapter, you enter all connection information into the Hive adapter datastore. You create the adapter in the Administrator module of the Management Console.

Complete the following group of tasks to connect to Hive using the Hive adapter:

1. Open the Administrator in the Management Console and enable the Job Server to support adapters.
2. In the Administrator, add, configure, and start an adapter instance.
3. In Data Services, add and configure a Hive adapter datastore.

i Note

Data Services supports Apache Hive and HiveServer2 version 0.11 and higher. For the most recent compatibility information, see the *Product Availability Matrix* (PAM) at <https://apps.support.sap.com/sap/support/pam>.

There are two types of Hive datastores:

- Hive adapter datastore: Data Services is installed within the Hadoop cluster. Configure the datastore without using DSN.
- Hive database datastore: Data Services is installed on any machine. Configure the datastore using DSN and a supported Hive ODBC driver.

Related Information

[Hive adapter datastore configuration options \[page 79\]](#)

[Using an ODBC driver to connect to a remote Hive Server \[page 82\]](#)

16 Reserved Words

16.1 About Reserved Words

The following words have special meanings in Data Services and therefore should not be used as names for work flows, data flows, transforms, or other design elements that you create. They should also not be used as user names when you create a Data Services repository. They are reserved with any combination of upper- and lower-case letters.

If you use reserved words you must put double quotation marks around them. For example:

```
"PRIMARY"
```

Reserved words appear in editor text areas in blue.

<code>_AL_DEFINE</code>	<code>_AL_ELSE</code>
<code>_AL_IFDEF</code>	<code>_AL_MESSAGE</code>
<code>_AL_METADATA_ELEMENT</code>	<code>_AL_STORED_PROCEDURE</code>
<code>_AL_TRAN_FUNCTION</code>	<code>_FUNC_TABLE</code>
<code>_MEMORY</code>	<code>_RFC_FUNCTION</code>
<code>_SAP_INNER_JOIN</code>	<code>_SAP_LEFT_OUTER_JOIN</code>
<code>ABAP_PROGRAM</code>	<code>ACTA</code>
<code>ACTAGUICOMMENT</code>	<code>ALGUICOMMENT</code>
<code>ALL</code>	<code>AL_NEST</code>
<code>AL_NESTED_TABLE</code>	<code>AL_PROJECT</code>
<code>AL_REAL_TIME_DATAFLOW</code>	<code>AL_RELATION</code>
<code>AL_REPO_FUNCTION</code>	<code>AL_RFC_SCHEMA_GROUP</code>
<code>AL_UNNEST</code>	<code>AL_UNNEST_SCHEMA_GROUP</code>
<code>AL_UNSPECIFIED_PARAMAND</code>	<code>AND</code>
<code>AS</code>	<code>ASC</code>
<code>BEGIN</code>	<code>BEGIN_SCRIPT</code>
<code>BULK</code>	<code>BY</code>

CALL	CASE
CATCH	CHAR
CHARACTER	CONCAT
CONVERT	CREATE
CUSTOM	
DATABASE	DATAFLOW
DATASTORE	DATE
DATETIME	DECIMAL
DECLARE	DEFAULT
DESC	DISTINCT
DISTINCT_KEY	DOMAIN
DOUBLE	
ELSE	EMBEDDED_DATAFLOW
EMBEDDED_DATAFLOW_RT	END
END_TRY	ERROR
ERROR_CONDITION	ERROR_STEP
FILE	FIRM_NOISE_WORD
FLOWOUTPUT	FOREIGN
FROM	FUNCTION
FUNC_ANY	FUNC_CHAR
FUNC_COL	FUNC_DS
FUNC_NUM	
GENERATED	GLOBAL
GROUP	
HAVING	
IF	IN
INPUT	INT
INTEGER	INTERVAL

IS	
JOB SERVER	KEY
LOCAL	
LEFTOUTERJOIN	LIKE
LOAD	LONG
LOOKUP	
MOD	
NOT	NULL
NUMERIC	
ON	OR
ORDER	OUT
OUTPUT	
PARALLEL	PIPE
PLAN	PRIMARY
PSFT_TREE	
READ	REAL
REFERENCES	RETURN
RETURNS	ROW
SAP_TREE	SELECT
SESSION	SET
SYSTEM	SYSTEM_PROFILE
TABLE	TIME
TRANSFORM	TRANSFORM_SCHEMA_MAPPING
TRY	
VARCHAR	VARIABLE
VIEW	VOID
WHERE	WHILE

17 Glossary

access server	A real-time, request-reply message broker that collects message requests, routes them to a real-time service, and delivers a message reply within the user-specified time.
adapter	An external Data Services interface that is created using the Software Developer Kit or is prebuilt and purchased from SAP.
Administrator	A browser-based application running on the Data Services Management Console that allows you to schedule, execute, monitor batch jobs, and so on.
after-image	The values in an UPDATE row after the row changes, used for log-based changed-data capture (CDC) jobs.
alias	An alternate form or name.
ANKLink	An NCOALink option that provides additional data about moves that occurred in the previous months 19 through 48.
annotation	A note that you can attach to a workspace diagram to describe or explain job, work flow, or data flow.
application	A software program.
association matching	A method that uses the results of two or more match sets to find matches that could not be found within a single match set.
audit point	The object in a data flow where audit statistics are collected.
batch job	A set of objects that you can schedule and execute together.
batch project	A job that executes at a specific time and ends after all the data in the specific source is processed.
before-image	The values in an UPDATE row before the row changes, used for log-based changed-data capture (CDC) jobs.
best record	A record created by consolidating the most complete, accurate, and up-to-date data elements from matching records.
best record priority	A value used to designate data from a particular source as having more or less importance than other data.
Binary Large Object	A field whose data consists of such objects as bitmap graphics, images, OLE objects, and metafiles. See also blob.
blank penalty	A setting that assigns a lower priority to records in which a particular field is blank.
blob	A field whose data consists of such objects as bitmap graphics, images, OLE objects, and metafiles. See also Binary Large Object.
blueprint	A sample job that has already been set up to handle a common business problem.
break group	A subset of records that are more likely to match, and which consist of driver and passenger records. Fields commonly used for creating break groups are postcodes,

	account or Social Security numbers, or the first two positions (or characters) of the street name.
break key	A user-defined field used to create a break group.
bulk loading	The moving of large amounts of data into a database to achieve optimal performance.
CASS	A USPS certification that requires software vendors to go through a series of tests to prove that their software correctly codes addresses according to USPS requirements, and produces the required USPS reports. See also Coding Accuracy Support System.
CDC	The process of retrieving changes made to a production data source. This process consolidates units of work, ensures, data is synchronized with the original source, and reduces data volume in a warehouse environment. See also changed-data capture.
CDC checkpoint	A software feature that lets you restrict changed-data capture (CDC) subscription reads.
CDC datastore	A datastore that allows you to limit extracted data to changed data only, and connects a changed-data capture (CDC) table in a source database to Data Services.
CDC subscription	An option on a source CDC table that defines the start and end of your data set, thereby allowing different data flows to extract data from the same table without corrupting data extracted by other data flows.
central repository	A storage mechanism that contains all information normally found in a local repository (definitions for each object in an application), but is optional and is shared by multiple users, who can check objects in and out of the repository.
changed-data capture	The process of retrieving changes made to a production data source. This process consolidates units of work, ensures, data is synchronized with the original source, and reduces data volume in a warehouse environment. See also CDC.
City directory	A file that is used by the USA Regulatory Address Cleanse transform when processing data from the U.S., and contains a table of city names, states, and ZIP Codes, organized by state and city.
classification	An indicator of the types of situations that apply to a word.
client/server	A distributed technology approach where the processing is divided by function. The server performs shared functions (such as managing communications and providing database services), while the client performs individual user functions.
Coding Accuracy Support System	A USPS certification that requires software vendors to go through a series of tests to prove that their software correctly codes addresses according to USPS requirements, and produces the required USPS reports. See also CASS.
Common Warehouse Model	A specification that enables interchange of data warehouse metadata between tools, platforms, and repositories in distributed heterogeneous environments. See also CWM.
compare buffer	A part of memory reserved for processing break groups (one break group at a time) in the Match or Associate transform. A larger buffer typically helps improve performance.
component	A major piece of the software.

conditional	A single-use object, available in work flows, that allows you to branch the execution logic based on the results of an expression. The conditional takes the form of an if/then/else statement.
connection string	A string version of the initialization properties needed to connect to a database, also known as a "DSN-less" connection. With a connection string you can easily store connection information or pass it between applications.
content type	The type of data in a field in your data source; helps you map your fields when you set up downstream transforms.
contribution value	A value you assign to a match criteria that represents the importance (or weight) you place on that criteria's data.
custom ABAP program	Software that extracts data from an SAP application using custom logic that is not currently supported by Data Services ABAP generation logic, and generates a data set that you use as a source in a data flow or an ABAP data flow.
custom function	A script you create to evaluate or make calculations on input values and produce a return value.
CWM	A specification that enables interchange of data warehouse metadata between tools, platforms, and repositories in distributed heterogeneous environments. See also Common Warehouse Model.
data collection	A collection of information that is sent between transforms.
data flow	A reusable object containing steps to define the transformation of information from source to target.
data record	A row of data that is constructed at run time.
Data Services engine	The core process that reads job information from the Data Services repository and sets up run-time processes that execute the job. The run-time processes extract, transform, and load relational and hierarchical data. The Job Server starts the Data Services engine to execute batch or real-time jobs.
Data Services repository	The database that contains information about a Data Services application. The repository contains information about defined reusable objects, and the metadata for sources and targets, transforms, and functions. The repository also contains the job history and runtime statistics information.
Data Services service	The process that ensures that the Access Server and the Job Server are running.
data set	Rows of data with a defined schema.
data source name	A parameter that provides connectivity for a Windows user to a database through an Open Database Connectivity (ODBC) driver. See also DSN.
data transformation	The phase of the data movement process that occurs between extraction and loading.
data transport	A step in an ABAP data flow that defines a target to store the data set extracted during the flow. You can locate the target file on the SAP Application server or in a location accessible to both the SAP Application server and to Data Services across a network.
data type	The format used to store a value, which can imply a default format for displaying and entering the value.

data validation	The process of defining rules to which correct data should conform. In Data Services you define these rules in the Validation transform.
database link	A communication path from one database server to another.
Database Management System	A software application that builds and maintains database tables. See also DBMS.
DataConnector	An operator instance used to read data files generated by Data Services when performing bulk loading using the Teradata Warehouse Builder.
datastore	A logical channel connecting Data Services to a source or target application.
datastore configuration	The definition of a connection to a particular database from a single datastore.
DBMS	A software application that builds and maintains database tables. See also Database Management System.
debug mode	A state of operation that allows you to diagnose errors while executing a job using the interactive debugging features in the Designer.
degree of parallelism	A property of a data flow that defines how many times each transform defined in the data flow replicates for use on a parallel subset of data. See also DOP.
delimiter	A character sequence used to separate column, row, and text data. To separate columns, a delimiter can be a tab, semicolon, comma, space, or any character sequence. To separate rows of data, a delimiter can be a {new line} or any other character sequence. To denote the start and end of a character string, a delimiter can be single quotation marks ('), double quotation marks ("), or {none}.
Delivery Point Barcode	A form of Postnet barcode, consisting of 62 bars and based on the combination of ZIP Code, ZIP+4, DPBC, and a check digit. See also DPBC.
Delivery Point Validation	A technology that assists you in validating the accuracy of your address information with the USA Regulatory Address Cleanse transform. With DPV you can identify addresses that are undeliverable as addressed and whether an address is a Commercial Mail Receiving Agency (CMRA). See also DPV.
Designer	A graphical user interface that allows you to design and test Data Services jobs.
diacritical character	A character that contains an accent, dieresis (umlaut), tilde, cedilla, or other distinguishing marks (for example, ä or Ç). You can choose to have standardized data with these types of characters. The application uses the Latin-1 code page for assigning these accents.
dictionary	Relational database that contains a lexicon of words and phrases that the data cleansing packages and the Data Cleanse transform use to identify, parse, and standardize data.
directional	A component of the address line that indicates direction, such as North in "211 North 115th St".
disambiguation	The process of resolving ambiguity.
discrete field	Input or output data that has separate fields for each piece of information, such as addresses and names.

discrete line format	Input source format in which pieces of data are parsed down to nearly the most distinct level. For example, a “first name” field would be discrete, whereas a “name” field that could contain first, middle, or last name information would not be discrete.
DOP	A property of a data flow that defines how many times each transform defined in the data flow replicates for use on a parallel subset of data. See also degree of parallelism.
DPBC	A form of Postnet barcode, consisting of 62 bars and based on the combination of ZIP Code, ZIP+4, DPBC, and a check digit. See also Delivery Point Barcode.
DPV	A technology that assists you in validating the accuracy of your address information with the USA Regulatory Address Cleanse transform. With DPV you can identify addresses that are undeliverable as addressed and whether an address is a Commercial Mail Receiving Agency (CMRA). See also Delivery Point Validation.
drill down	To explore detailed data that was used in creating a summary level of data. How far you drill down depends on the granularity of the data in the warehouse.
driver record	A record that drives the comparison process. Driver records are part of a break group and are compared with passenger records to determine matches.
DSN	A parameter that provides connectivity for a Windows user to a database through an Open Database Connectivity (ODBC) driver. See also data source name.
dual address	A dual address occurs when a record contains two address lines. Two combinations are typical: • PO box and street address: 1000 Main Street, Suite 51 / PO Box 2342 • Rural route or Highway Contract and street address: RR 1 Box 345 / 12784 Old Columbus Road
dual names	Two names included on an address line, such as John and Jane Doe.
Early Warning System	A solution for matching valid delivery points that have been created between updates to the national ZIP+4 directory. EWS uses four months of rolling data found in an intermediate directory that is updated weekly with data from the USPS. See also EWS.
eLOT	A sorting sequence for US mail in which ZIP+4 codes are arranged in the order that they are served by the mail carrier. Compare with Line of Travel (LOT). See also Enhanced Line of Travel.
embedded data flow	A data flow that is called from inside another data flow.
Enhanced Line of Travel	A sorting sequence for US mail in which ZIP+4 codes are arranged in the order that they are served by the mail carrier. Compare with Line of Travel (LOT). See also eLOT.
enterprise application	Software that enables businesses to execute and optimize business and IT strategies in domains like Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), or Supply Chain Management (SCM).
enterprise resource planning system	An enterprise application from which Data Services can extract data. See also ERP system.
ERP system	An enterprise application from which Data Services can extract data. See also enterprise resource planning system.
EWS	A solution for matching valid delivery points that have been created between updates to the national ZIP+4 directory. EWS uses four months of rolling data found in an intermediate directory that is updated weekly with data from the USPS. See also Early Warning System.

exception	An error that occurs while executing a job.
expression	A combination of variables, parameters, constants, and functions linked by operation symbols and any required punctuation that describe a rule for calculating a value.
fault code	A numeric value that is assigned to a record after the USA Regulatory Address Cleanse transform validation process that signifies that the particular record was not successfully validated. Each numeric value represent a different type of fault.
file format	A description of how data is or should be organized in a file Data Services reads from or loads to. A file format can be specific to a single file or generic for many files.
Forward Sortation Area	The first three characters of a Canadian alphanumeric postal code, such as K1A in the postal code for Canada Post's Ottawa headquarters, K1A 0B1. See also FSA.
FSA	The first three characters of a Canadian alphanumeric postal code, such as K1A in the postal code for Canada Post's Ottawa headquarters, K1A 0B1. See also Forward Sortation Area.
function	A program that operates on values that are passed to it.
functional area	A virtual group of jobs that relate to the same business function, such as Human Resources or Customers.
gather	To recombine terms, such as alphanumeric terms that you would look up together in the dictionary.
gender	A code that indicates the likelihood of a record being a certain sex. This code is derived from the name and has five possible values: strong male, strong female, weak male, weak female, ambiguous, and unassigned.
generated field	A field that is produced on output by a transform, such as a postcode field generated by the Global Address Cleanse transform.
global suggestion list	A method of completing and populating addresses with minimal data, or offering suggestions for possible matches. This address-entry system is ideal in call center environments or any transactional environment where data cleansing is necessary at the point of entry, and a human being is available to choose one of the suggestions.
host name	A computer's network name (or IP address).
hybrid format	An arrangement for records in which some fields are discrete whereas others are in a multiline format.
impact and lineage analysis	A category of reports on the Management Console that provides end-to-end impact and lineage analysis of Data Services tables and columns and SAP BusinessObjects Enterprise objects such as business views and reports.
import	The process of acquiring information for the Data Services repository.
input source	The records in a database or file that you want the application to process.
interactive debugger	A Designer feature that allows you to step through the data of a job one row at a time using filters and breakpoints on a line.
interface	A type of interaction with Data Services which is either internal (allows you to create datastore connections to natively- supported applications), or external (allows Data Services to to communicate with information exchange technologies such as Web Services and MQ queries).

intersource match	A match between records of different sources.
job	The unit of work that can be scheduled independently for execution by the Administrator. Jobs are special work flows that can be scheduled for execution, but cannot be called by other work flows or jobs.
Job Server	The Data Services software that receives requests from the Designer and the Administrator to start and stop jobs.
join rank	A value given to or calculated for all data sets in a data flow by which the application determines which source to read first when assembling the data set in a join.
LACS	A database of addresses that have been permanently converted, usually due to 911 emergency system implementation. The changes often consist of conversion from rural-style addressing to standardized, city-style addressing, or renumbering of existing city-style addresses. See also Locatable Address Conversion System.
lastline	The address information that contains components such as the locality, region, postcode, and sometimes country name.
LDU	The last three characters of a Canadian alphanumeric postal code. Compare with Forward Sortation Area (FSA). See also Local Delivery Unit.
legacy system	An information or transaction processing system used to store data such as bank balances, inventories, payroll, and manufacturing parts.
license-controlled feature	A feature that is enabled or disabled depending on the product license.
line	The visual connection between objects in a workspace diagram which shows the left-to-right flow path for data moving between those objects during job execution.
Line of Travel	A sorting sequence for US mail in which ZIP+4 codes are arranged in the order that they are served by the mail carrier. See also LOT.
linked datastores	The datastores in a database link relationship.
Local Delivery Unit	The last three characters of a Canadian alphanumeric postal code. Compare with Forward Sortation Area (FSA). See also LDU.
locale	The identification of a record's world region, which controls the format of data when it is stored, processed, or displayed.
locality	A part of the address line of a record that usually refers to the city or town, but which in some countries, such as the United Kingdom, can extend to include the district.
Locatable Address Conversion System	A database of addresses that have been permanently converted, usually due to 911 emergency system implementation. The changes often consist of conversion from rural-style addressing to standardized, city-style addressing, or renumbering of existing city-style addresses. See also LACS.
lookup table	A table that contains data that other tables can reference with lookup functions that return one or more output columns.
LOT	A sorting sequence for US mail in which ZIP+4 codes are arranged in the order that they are served by the mail carrier. See also Line of Travel.
mail piece unit	A version identifier for printers, representing the unique characteristics of a portion of a mailing.

mapped field	A data-quality-specific field used to tell a data quality transform how to process the data in that field.
master record	The first record in a match group.
match	A pair or group of records that are found to be identical, based on the criteria you set.
match criteria	The options and rules you define for how a match key is used to find records in your data.
match group	A collection of records, consisting of a master and subordinate records, that are found to be matching with each other.
match set	A group of options and rules used to perform comparisons on data.
memory datastore	A datastore connection to an in-memory database.
memory table	An internal table used to store a data set in memory while a job runs.
Meta Integration Model Bridge	A Windows stand-alone utility that converts metadata models among design tool formats. See also MIMB.
metadata	Information acquired and maintained to describe tables in source and target databases.
Metropolitan Statistical Area	An aggregation of counties by the US Office of Management and Budget used for statistical purposes. See also MSA.
MIMB	A Windows stand-alone utility that converts metadata models among design tool formats. See also Meta Integration Model Bridge.
MSA	An aggregation of counties by the US Office of Management and Budget used for statistical purposes. See also Metropolitan Statistical Area.
multiline	A database record format in which address data is not consistently located in the same arrangement in all records.
multiline field	Input or output data that has certain address and name data combined in one field.
NANP	A system for structuring telephone numbers that is shared by 19 North American countries including the United States and Canada. See also North American Numbering Plan.
NCOALink	A USPS product consisting of a secured database of approximately 160 million change of address (COA) records consisting of the names and addresses of individuals, families, and businesses who have filed a change of address with the USPS.
nested data	Information in one table that is related to a single row in another table.
noise word	A word that is meaningless to the matching process.
normal source	An origin of records that the application should consider to be good, eligible records.
North American Numbering Plan	A system for structuring telephone numbers that is shared by 19 North American countries including the United States and Canada. See also NANP.
object	Any item that you create in the Data Services Designer.
object definition	The options that describe the operation of an object, which are viewable in the workspace when you open the object.

object dependent	The state of being associated beneath another object. Any object under the highest level object in the hierarchy is object dependent.
object library	A directory management system that provides access to reusable objects.
object version	An instance of an object. Each time you add or check in an object to the central repository, Data Services creates a new version of the object.
operation code	A flag associated with a row in a data set that indicates the status of the data in the row, such as INSERT, UPDATE, DELETE, and NORMAL.
operational dashboard	A type of report on the Management Console that visually displays the status and performance of job and data flow executions.
Option Editor	A feature in Data Services' Data Quality transform editor through which you can change the value for each option within the transform.
Option Explorer	A pane in the Associate, Match, and User-Defined transform editors which shows a list of the option groups within a transform.
option group	A set of choices that control various business rules for a transform.
other source	In a Match transform, data that should be treated as transparent, such as seed sources, and as such are not counted in determining how to characterize a match group.
parameter	A value passed to a work flow or data flow when that flow is called.
partition	The division of table data into sets based on criteria such as a range or list of values in each row.
passenger record	A row of data in a break group that is compared against the driver record.
pattern file	A plain text file that contains user-defined patterns and is used by the Data Cleanse transform, and can be edited by any text editing program.
PMB	A postal delivery location similar to a post-office box but which is hosted by a private company. See also Private Mail Box.
postal code	A system of letters and/or digits used for sorting mail, such as the numeric ZIP Code used in the U.S. and the alphanumeric FSA LDU system used in Canada.
postcode move	A valid postcode that has been split or moved, so only a portion of the the area that had been covered by the one postcode now has two or more postcodes, including the original one, for the same area.
postcode1	The postal code or five-digit ZIP Code (USA).
postcode2	The secondary part of a postal code, such as the "4051" in the United States postcode "54601-4051".
primary entry	A word or phrase in the dictionary that the data cleansing packages and Data Cleanse transform use to identify, parse, and standardize data.
Private Mail Box	A postal delivery location similar to a post-office box but which is hosted by a private company. See also PMB.
project	The highest-level object in the Designer window, which provides you with a way to organize the other objects you create in Data Services.

projection	An operation within a SELECT statement that the software can push to the database; the subset of columns that you map on the Mapping tag in the query editor.
property	An item of information that describes an object, such as its name, description, or date on which it was created.
query transform	A data transformation object that creates a data set that satisfies conditions you specify.
real-time job	A job that executes on-demand as a "request-response" system.
reference file	A file of address data that Data Services can use to match, assign, standardize, and verify addresses.
relational data	A data set in which data in each column contains a scalar value.
Remote Function Call server	A server that allows third-party programs, including SAP Applications and SAP NetWeaver Business Warehouse, to schedule and initiate Data Services jobs and return the results to Data Services. See also RFC server.
Remote Function Call server interface	The node on the Administrator application of the Data Services Management Console where you configure SAP connections to load data into or read data from an SAP NetWeaver Business Warehouse system. See also RFC server interface.
repository	A set of tables that hold user-created and predefined system objects, source and target metadata, and transformation rules.
request/acknowledge operation	An operation that executes a remote HTTP service in the Request Acknowledge mode, wherein acknowledgement is sent only if the operation is successful.
request/reply operation	An operation that sends a request and then awaits notice of the request's result.
reusable object	An object that can be defined, stored, and reused independent of other objects, and is accessible from the object library.
RFC server	A server that allows third-party programs, including SAP Applications and SAP NetWeaver Business Warehouse, to schedule and initiate Data Services jobs and return the results to Data Services. See also Remote Function Call server.
RFC server interface	The node on the Administrator application of the Data Services Management Console where you configure SAP connections to load data into or read data from an SAP NetWeaver Business Warehouse system. See also Remote Function Call server interface.
rule file	A text file that controls how the application parses data.
rule matching	The process of comparing token classifications against defined rules.
sample size	The number of rows to display in the View Data feature.
sampling rate	The number of rows processed after which Data Services writes information to the monitor log file and updates job events.
sampling rows	The parameter that specifies the frequency with which the Management Console Profiler samples data, beginning with the first row of the specified number of sampling rows.

script	A step in a job or work flow that allows you to calculate values to pass to other parts of the job or work flow by calling functions, executing if-then-else statements, and assigning values to variables.
secondary information	Data that helps the application determine how to process a string in various scenarios.
segment	The format with which the data records of IDocs are interpreted.
server group	An association of Job Servers on different computers that can automatically measure resource availability, and distribute batch jobs or part of a job to the Job Server with the lightest load at run time.
similarity score	A percentage that indicates how much two fields or values are considered alike, which is calculated by the application after the comparison process.
single use object	An object that is defined only within the context of one job or one data flow.
smart editor	A flexible Data Services tool used for creating scripts, expressions, and custom functions without having to type the names of existing elements like column, function, and variable names.
snowbird	An informal term for a person with multiple residences who typically changes where he or she resides according to the season.
source group	A collection of data that you can use to prepare a second set of match statistics, combining the statistics for two or more regular sources.
source record	A row that contains the data you want to use for updating or creating your best record.
standards	Business rules that define how Data Cleanse will apply capitalization or other output formatting to data.
star schema	A database design used to format data in a data mart, and which is based on a single fact table to which any number of dimensional tables may be joined.
step	An object that is part of the definition of a work or data flow, which is represented by an icon in the flow diagram, and is connected to other steps to indicate the flow of data through the data flow, or the order of execution in the work flow.
street address	A postal delivery location that consists of a street name and house number.
subordinate record	Any record in a match group other than the master record.
substitution parameter	A text string "alias" that you can use within your job and transforms, and is defined in a substitution parameter configuration. At runtime, that parameter is replaced with its value anywhere it is used in your job.
substitution parameter configuration	The definition of the substitution parameters used throughout your job in a particular run-time environment.
suggestion list	A group of potential matches presented to the user for selection of the correct one.
suppression source	An origin of data that contains records of information that should be excluded from other output destinations.
table	Database information that is organized into rows and columns that the software reads data from or loads data into.



target	The object into which the application loads extracted and transformed data in a data flow.
TDPID	The server name Data Services uses when loading with the bulk loader option. See also Teradata Director Program ID.
Teradata Director Program ID	The server name Data Services uses when loading with the bulk loader option. See also TDPID.
territory	The locale value for a geographical location (usually the country) where a local language is used.
thread	The instance of the program running on behalf of a process.
tokenization	The creation of tokens, which assigns meaning to each piece of word that results from hyphenation in the Data Cleanse dictionary.
transform	A step in a data flow that acts on a data set, and is available through the object library in three categories: Data Integrator, Data Quality, and Platform.
try/catch block	A combination of a try object and one or more catch objects that define alternate execution paths in case an error occurs during the execution of a job.
unique identifier	In a Data Quality transform, an ID that is unique to a record or group of matching records, and is sequential, static, and will not change when records are updated or re-processed through the application.
unique record	A record that does not have any matching or subordinate records and so does not belong to any match group after the matching process is complete.
web service request	Any message sent from a web client that requires processing by a real-time job.
web services	A platform on which multiple applications can communicate with each other even though constructed in different languages and on different platforms.
weighted scoring	A method of comparison that lets you use values to place more or less importance on various match criteria during the matching process.
work flow	A reusable object that contains steps defining the order of job execution.
workspace	The area on the Designer window where you can manipulate system objects and graphically assemble data movement processes.
Z4Change	A directory of all U.S. ZIP Codes and ZIP+4 Codes, in which those codes that have changed in the last 12 months are flagged.
ZCF	A directory that is used by the USA Regulatory Address Cleanse transform when processing data from the U.S., and contains a table of city names, states, and ZIP Codes, organized by ZIP Code. See also ZIP City File.
ZIP City File	A directory that is used by the USA Regulatory Address Cleanse transform when processing data from the U.S., and contains a table of city names, states, and ZIP Codes, organized by ZIP Code. See also ZCF.
ZIP plus 4	A postal code that consists of both the USPS's 5-digit ZIP Code and the 4-digit add-on code.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.

About the icons:

- Links with the icon  : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
 - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
 - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon  : You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.

The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

© 2018 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.