PUBLIC

Document Version: 4.2 Support Package 14 (14.2.14.0) – 2022-09-17

Data Services Supplement for Big Data

THE BEST RUN SAP

Content

1 About this supplement

This supplement contains information about the big data products that SAP Data Services supports.

The supplement contains information about the following:

- Supported big data products
- Supported cloud computing technologies including cloud databases and cloud storages.

Find basic information in the *Reference Guide*, *Designer Guide*, and some of the applicable supplement guides. For example, to learn about datastores and creating datastores, see the *Reference Guide*. To learn about Google BigQuery, refer to the *Supplement for Google BigQuery*.

2 Naming conventions and variables

This documentation uses specific terminology, location variables, and environment variables that describe various features, processes, and locations in SAP Business Objects and SAP Data Services.

Terminology

SAP Data Services documentation uses the following terminology:

- The terms **Data Services system** and **SAP Data Services** mean the same thing.
- The term **BI platform** refers to **SAP BusinessObjects Business Intelligence platform**.
- The term **IPS** refers to **SAP BusinessObjects Information platform services**.

  > **i** Note
  >
  > Data Services requires BI platform components. However, when you don't use other SAP applications, IPS, a scaled back version of BI, also provides these components for Data Services.

- **CMC** refers to the Central Management Console provided by the BI or IPS platform.
- **CMS** refers to the Central Management Server provided by the BI or IPS platform.

Variables

The following table describes the location variables and environment variables that are necessary when you install and configure Data Services and required components.

Variables	Description
INSTALL_DIR	The installation directory for SAP applications such as Data Services. Default location: • For Windows: C:\Program Files (x86)\SAP BusinessObjects • For UNIX: $HOME/sap businessobjects **Note** INSTALL_DIR isn't an environment variable. The installation location of SAP software can be different than what we list for INSTALL_DIR based on the location that your administrator sets during installation.
BIP_INSTALL_DIR	The directory for the BI or IPS platform. Default location: • For Windows: <INSTALL_DIR>\SAP BusinessObjects Enterprise XI 4.0 **Example** C:\Program Files (x86)\SAP BusinessObjects\SAP BusinessObjects Enterprise XI 4.0 • For UNIX: <INSTALL_DIR>/enterprise_xi40 **Note** These paths are the same for both BI and IPS. **Note** BIP_INSTALL_DIR isn't an environment variable. The installation location of SAP software can be different than what we list for BIP_INSTALL_DIR based on the location that your administrator sets during installation.

Variables	Description
<LINK_DIR>	An environment variable for the root directory of the Data Services system. Default location: • All platforms <LINK_DIR>\Data Services **Example** C:\Program Files (x86)\SAP BusinessObjects\Data Services

Variables	Description
<DS_COMMON_DIR>	An environment variable for the common configuration directory for the Data Services system. Default location: • If your system is on Windows (Vista and newer): <AllUsersProfile>\SAP BusinessObjects\Data Services **i Note** The default value of <AllUsersProfile> environment variable for Windows Vista and newer is C:\ProgramData. **Example** C:\ProgramData\SAP BusinessObjects\Data Services • If your system is on Windows (Older versions such as XP) <AllUsersProfile>\Application Data\SAP BusinessObjects\Data Services **i Note** The default value of <AllUsersProfile> environment variable for Windows older versions is C:\Documents and Settings\All Users. **Example** C:\Documents and Settings\All Users\Application Data\SAP BusinessObjects\Data Services • UNIX systems (for compatibility) <LINK_DIR> The installer automatically creates this system environment variable during installation. **i Note** Starting with Data Services 4.2 SP6, users can designate a different default location for <DS_COMMON_DIR> during installation. If you can't find the <DS_COMMON_DIR> in the listed default location, ask

Variables	Description
	your System Administrator to find out where your default location is for <DS_COMMON_DIR>.
<DS_USER_DIR>	The environment variable for the user-specific configuration directory for the Data Services system. Default location: • If you're on Windows (Vista and newer): <UserProfile>\AppData\Local\SAP BusinessObjects\Data Services **Note** The default value of <UserProfile> environment variable for Windows Vista and newer versions is C:\Users\{username}. • If you're on Windows (Older versions such as XP): <UserProfile>\Local Settings\Application Data\SAP BusinessObjects\Data Services **Note** The default value of <UserProfile> environment variable for Windows older versions is C:\Documents and Settings\{username}. **Note** The system uses <DS_USER_DIR> only for Data Services client applications on Windows. UNIX platforms don't use <DS_USER_DIR>. The installer automatically creates this system environment variable during installation.

3 Big data in SAP Data Services

SAP Data Services supports many types of big data through various object types and file formats.

3.1 Apache Cassandra

Apache Cassandra is an open-source data storage system that you can access with SAP Data Services.

Data Services natively supports Cassandra as an ODBC data source with a DSN connection. Cassandra uses the generic ODBC driver. Use Cassandra on Windows or Linux operating systems.

Use Cassandra data for the following tasks:

- Use as sources, targets, or template tables
- Preview data
- Query using `distinct`, `where`, `group by`, and `order by`
- Write scripts using functions such as `math`, `string`, `date`, `aggregate`, and `ifthenelse`

Before you use Cassandra with Data Services, ensure that you perform the following setup tasks:

- Add the appropriate environment variables to the `al_env.sh` file.
- For Data Services on Linux platforms, configure the ODBC driver using the Connection Manager.

> **i Note**
>
> For Data Services on Windows platforms, use the generic ODBC driver.

For more information about configuring database connectivity for UNIX and Linux, see the *Administrator Guide*.

3.1.1 Setting ODBC driver configuration on Linux

Use the Connection Manager to configure the ODBC driver for Apache Cassandra on Linux.

Before you complete the following steps, read the topic and subtopics under "Configure database connectivity for UNIX and Linux" in the *Administrator Guide*.

Use the GTK+2 library to make a graphical user interface for the Connection Manager. Connection Manager is a command-line utility. To use it with a UI, install the GTK+2 library. For more information about obtaining and installing GTK+2, see https://www.gtk.org/ . The following steps are for the UI for Connection Manager.

1. Open a command prompt and set `$ODBCINI` to a file in which the Connection Manager defines the DSN. Ensure that the file is readable and writable.

   > **Sample Code**
   >
   > ```
   > $ export ODBCINI=<dir-path>/odbc.ini
   > touch $ODBCINI
   > ```

   The Connection Manager uses the `$ODBCINI` file and other information that you enter for data sources, to define the DSN for Cassandra.

   > **i Note**
   >
   > Do not point to the Data Services ODBC `.ini` file.

2. Start the Connection Manager user interface by entering the following command:

   Sample Code

   ```
   $ cd <LINK_DIR>/bin/
   $ /DSConnectionManager.sh
   ```

   Note

   <LINK_DIR> is the Data Services installation directory.

3. In Connection Manager, open the *Data Sources* tab, and click *Add* to display the list of database types.
4. In the *Select Database Type* dialog box, select Cassandra and click *OK*.

   The *Configuration for...* dialog box opens. It contains the absolute location of the `odbc.ini` file that you set in the first step.
5. Provide values for additional connection properties for the Cassandra database type as applicable.
6. Provide the following properties:

   - *User name*
   - *Password*

   Note

   Data Services does not save these properties for other users.

7. To test the connection, click *Test Connection*.
8. Click *Restart Services* to restart services applicable to the Data Services installation location:

   If Data Services is installed on the same machine and in the same folder as the IPS or BI platform, restart the following services:

   - EIM Adaptive Process Service
   - Data Services Job Service

   If Data Services is not installed on the same machine and in the same folder as the IPS or BI platform, restart the following service:

   - Data Services Job Service

3.1.2 Data source properties for Cassandra

Complete data source properties in the Connection Manager when you configure the ODBC driver for SAP Data Services on Linux.

The Connection Manager configures the `$ODBCINI` file based on the property values that you enter in the *Data Sources* tab. The following table lists the properties that are relevant for Apache Cassandra.

Data Source settings for Apache Cassandra

Database Type	Properties on Data Sources tab
Apache Cassandra	• *User Name* • *Database password* • *Host Name* • *Port* • *Database* • *Unix ODBC Lib Path* • *Driver* • *Cassandra SSL Certificate Mode [0:disabled|1:one-way|2:two-way]* Depending on the value you choose for the certificate mode, Data Services may require you to define some or all of the following options: • *Cassandra SSL Server Certificate File* • *Cassandra SSL Client Certificate File* • *Cassandra SSL Client Key File* • *Cassandra SSL Client Key Password* • *Cassandra SSL Validate Server Hostname? [0:disabled|1:enabled]*

3.2 Apache Hadoop

Use SAP Data Services to connect to Apache Hadoop frameworks including Hadoop Distributive File Systems (HDFS) and Hive.

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) on the SAP Support Portal for the supported versions of HDP.

For information about deploying Data Services on a Hadoop MapR cluster machine, see SAP Note 2404486.

For information about accessing your Hadoop in the administered SAP Big Data Services, see the *Supplement for SAP Big Data Services*.

For complete information about how Data Services supports Apache Hadoop, see the *Supplement for Hadoop*.

3.3 Hadoop Distributed File System (HDFS)

Connect to your HDFS data using an HDFS file format or an HDFS file location.

Create an HDFS file format and file location with your HDFS connection information, including account name, password, security protocol, and so on. SAP Data Services uses this information to access HDFS data during Data Services processing.

For complete information about how Data Services supports your HDFS, see the *Supplement for Hadoop*.

3.4 Hadoop Hive

Use a Hive adapter datastore or a Hive database datastore to connect to the Hive remote server.

Use the Hive adapter datastore when SAP Data Services is installed within the Hadoop cluster. Use the Hive adapter datastore for server-named (DSN-less) connections. Also include SSL (or the newer Transport Layer Security TLS) for secure communication over the network.

Use a Hive database datastore when Data Services is installed on a machine either within the Hadoop cluster or not. Use the Hive database datastore for a DSN or a DSN-less connection. Also include SSL (or the newer TLS) for secure communication over the network.

For complete information about how Data Services supports Hadoop Hive, see the *Supplement for Hadoop*.

3.5 Upload data to HDFS in the cloud

Upload data processed with Data Services to your HDFS that is managed by SAP Big Data Services.

Big Data Services is a Hadoop distribution in the cloud. Big Data Services performs all Hadoop upgrades and patches for you and provides Hadoop support. SAP Big Data Services was formerly known as Altiscale.

Upload your big data files directly from your computer to Big Data Services. Or, upload your big data files from your computer to an established cloud account, and then to Big Data Services.

> **Example**
>
> Access data from S3 (Amazon Simple Storage Service) and use the data as a source in Data Services. Then upload the data to your HDFS that resides in Big Data Service in the cloud.

How you choose to upload your data is based on your use case.

For complete information about accessing your Hadoop account in Big Data Services and uploading big data, see the *Supplement for SAP Big Data Services*.

3.6 Google Cloud Dataproc clusters

To connect to an Apache Hadoop web interface running on Google Cloud Dataproc clusters, use a Hive database datastore and a WebHDFS file location.

Use a Hive datastore to browse and view metadata from Hadoop and to import metadata for use in data flows. To upload processed data, use a Hadoop file location and a Hive template table. Implement bulk loading in the target editor in a data flow where you use the Hive template table as a target.

For complete information about how SAP Data Services supports Google Cloud Dataproc clusters, see the *Supplement for Hadoop*.

3.7 Apache Impala

Apache Impala is an open source database for Apache Hadoop.

To work with data from Apache Impala in SAP Data Services, import the Cloudera driver and configure a data source name (DSN) connection. Then create an Apache Impala datastore.

Data Services supports Impala 2.5 and later versions. Data Services supports only Impala scalar data types, but doesn't support complex types such as ARRAY, STRUCT, or MAP.

For complete information about how Data Services supports Apache Impala, see the *Suppplement for Hadoop*.

Related Information

Apache Impala [page 41]

3.8 HP Vertica

Access HP Vertica data in SAP Data Services by creating an HP Vertica database datastore.

Use HP Vertica data as sources or targets in data flows. Implement SSL secure data transfer with MIT Kerberos to access HP Vertica data securely. Additionally, configure options in the source or target table editors to enhance HP Vertica performance.

3.8.1 Enable MIT Kerberos for HP Vertica SSL protocol

SAP Data Services uses MIT Kerberos 5 authentication to securely access an HP Vertica database using SSL protocol.

You must have Database Administrator permissions to install MIT Kerberos 5 on your Data Services client machine. Additionally, the Database Administrator must establish a Kerberos Key Distribution Center (KDC) server for authentication. The KDC server must support Kerberos 5 using the Generic Security Service (GSS) API. The GSS API also supports non_MIT Kerberos implementations, such as Java and Windows clients.

> **i** Note
>
> Specific Kerberos and HP Vertica database processes are required before you can enable SSL protocol in Data Services. For complete explanations and processes for security and authentication, consult your HP Vertica user documentation and the MIT Kerberos user documentation.

MIT Kerberos authorizes connections to the HP Vertica database using a ticket system. The ticket system eliminates the need for users to enter a password.

Parent topic: HP Vertica [page 15]

Related Information

Creating a DSN for HP Vertica with Kerberos SSL [page 19]
Creating HP Vertica datastore with SSL encryption [page 21]
Increasing loading speed for HP Vertica [page 23]
HP Vertica data type conversion [page 23]
HP Vertica table source [page 25]
HP Vertica target table configuration [page 26]
Information to edit configuration or initialization file [page 16]
Generate secure key with kinit command [page 18]
Creating a DSN for HP Vertica with Kerberos SSL [page 19]

3.8.1.1 Information to edit configuration or initialization file

Descriptions for kerberos properties for configuration or initialization files.

After you install MIT Kerberos, define the specific Kerberos properties in the Kerberos configuration or initialization file and save it to your domain. For example, save `krb5.ini` to `C:\Windows`.

See the MIT Kerberos documentation for information about completing the Unix `krb5.conf` property file or the Windows `krb5.ini` property file. Kerberos documentation is located at: http://web.mit.edu/kerberos/krb5-current/doc/admin/conf_files/krb5_conf.html .

Log file locations for Kerberos

[logging] Locations for Kerberos log files

Property	Description
default = <value>	The location for the Kerberos library log file, `krb5libs.log`. For example: `default = FILE:/var/log/krb5libs.log`
kdc = <value>	The location for the Kerberos Data Center log file, `krb5kdc.log`. For example: `kdc = FILE:/var/log/krb5kdc.log`
admin_server = <value>	The location for the administrator log file, `kadmind.log`. For example: `admin_server = FILE:/var/log/kadmind.log`

Kerberos 5 library settings

[libdefaults] Settings used by the Kerberos 5 library

Property	Description
default_realm = <value>	The location of your domain. Example: `default_realm = EXAMPLE.COM` Domain must be in all capital letters.
dns_lookup_realm = <value>	Set to False: `dns_lookup_realm = false`
dns_lookup_kdc = <value>	Set to False: `dns_lookup_kdc = false`
ticket_lifetime = <value>	Set number of hours for the initial ticket request. For example: `ticket_lifetime = 24h` The default is 24h.
renew_lifetime = <value>	Set number of days a ticket can be renewed after the ticket lifetime expiration. For example: `renew_lifetime = 7d` The default is 0.
forwardable = <value>	Initial tickets can be forwarded when this value is set to True. For example: `forwardable = true`

Kerberos realm values

[realms] Value for each Kerberos realm

Property	Description
<kerberos_realm> = {<subsection_property = value>}	Location for each property of the Kerberos realm. For example: EXAMPLE.COM = {kdc=<location> admin_server=<location> kpasswd_server=<location>} Properties include: • KDC location • Admin Server location • Kerberos Password Server location **Note** Host and server names are lowercase.

Kerberos domain realm

[domain_realm]

Property	Description
<server_host_name>=<kerberos_realm>	Maps the server host name to the Kerberos realm name. If you use a domain name, prefix the name with a period (.).

Related Information

Generate secure key with kinit command [page 18]

3.8.1.2 Generate secure key with kinit command

Execute the `kinit` command to generate a secure key.

After you have updated the configuration or initialization file and saved it to the client domain, execute the `kinit` command to generate a secure key.

For example, enter the following command using your own information for the variables: `kinit <user_name>@<realm_name>`

The command should generate the following keys:

Key	Description
`-k`	Precedes the service name portion of the Kerberos principal. The default is `vertica`.
`-K`	Precedes the instance or host name portion of the Kerberos principal.
`-h`	Precedes the machine host name for the server.
`-d`	Precedes the HP Vertica database name that you want to connect to.
`-U`	Precedes the user name of the administrator user.

See the MIT Kerberos ticket management documentation for complete information about using the `kinit` command to obtain tickets: http://web.mit.edu/kerberos/krb5-current/doc/user/tkt_mgmt.html .

3.8.2 Creating a DSN for HP Vertica with Kerberos SSL

To enable SSL for HP Vertica database datastores, first create a data source name (DSN).

This procedure is for HP Vertica users who have database administrator permissions to perform these steps. Other non-database administrators may access the HP Vertica database only when they are associated with an authentication method through a GRANT statement.

To create a DSN for HP Vertica, use SAP Data Services 4.2 SP7 Patch 1 (14.2.7.1) or later version.

Install MIT Kerberos 5 and perform all of the required steps for MIT Kerberos authentication for HP Vertica. See your HP Vertica documentation in the security and authentication sections for details.

1. Open the ODBC Data Source Administrator.

   Access the ODBC Data Source Administrator either from the *Datastore Editor* in Data Services Designer or directly from your *Start* menu.
2. In the ODBC Data Source Administrator, open the *System DSN* tab and click *Add*.
3. Select the applicable HP Vertica driver from the list and click *Finish*.
4. Open the *Basic Settings* tab and complete the following options:

   HP Vertica ODBC DSN Configuration Basic Settings tab

Option	Value
DSN	Enter the HP Vertica data source name.
Description	Optional. Enter a description for this data source.

Option	Value
Database	Enter the name of the database that is running on the server.
Server	Enter the server name.
Port	Enter the port number on which HP Vertica listens for ODBC connections. The default is 5433.
User Name	Enter the database user name. The database user has DBADMIN permission, or is associated with the authentication method through a GRANT statement.

5. Optional. Select *Test Connection*.

> **i Note**
>
> If the connection fails, either continue with the configuration and fix the connection issue later, or reconfigure the connection information and try to test the connection again.

6. Open the *Client Settings* tab and complete the options as described in the following table.

HP Vertica ODBC DSN Configuration Client Settings tab

Option	Value
Kerberos Host Name	Enter the name of the host computer where Kerberos is installed.
Kerberos Service Name	Enter the applicable value.
SSL Mode	Select *Require*.
Address Family Preference	Select *None*.
Autocommit	Select this option.
Driver String Conversions	Select *Output*.
Result Buffer Size (bytes)	Enter the applicable value in bytes. Default is 131072.
Three Part Naming	Select this option.
Log Level	Select *No logging* from the dropdown list.

7. Click *Test Connection*. When the connection test is successful, click *OK* and close the ODBC Data Source Administrator.

   Now the HP Vertica DSN that you just created is included in the DSN option in the datastore editor.

Create the HP Vertica database datastore in Data Services Designer and select the DSN that you just created.

Task overview: HP Vertica [page 15]

Related Information

Enable MIT Kerberos for HP Vertica SSL protocol [page 16]
Creating HP Vertica datastore with SSL encryption [page 21]
Increasing loading speed for HP Vertica [page 23]
HP Vertica data type conversion [page 23]
HP Vertica table source [page 25]
HP Vertica target table configuration [page 26]
Creating HP Vertica datastore with SSL encryption [page 21]

3.8.3 Creating HP Vertica datastore with SSL encryption

SSL encryption protects data as it transfers between the database server and Data Services.

An administrator must install MIT Kerberos 5 and enable Kerberos for HP Vertica SSL protocol. Additionally, an administrator must create an SSL data source name (DSN) using the ODBC Data Source Administrator. Then the DSN is available to choose when you create the datastore. See the *Administrator Guide* for more information about configuring MIT Kerberos.

SSL encryption for HP Vertica is available in SAP Data Services version 4.2 Support Package 7 Patch 1 (14.2.7.1) or later.

> **i** Note
>
> Enabling SSL encryption slows down job performance.

> **i** Note
>
> An HP Vertica database datastore requires that you choose DSN as a connection method. DSN-less connections are not allowed for HP Vertica datastore with SSL encryption.

1. In Designer, select *Project* > *New* > *Datastore*.
2. Complete the options as you would for an HP Vertica database datastore. Complete the following options specifically for SSL encryption:

   SSL-specific options

Option	Value
Use Data Source Name (DSN)	Select this option.
Data Source Name	Select the HP Vertica SSL DSN data source file that was created previously in the ODBC Data Source Administrator.

3. Complete the remaining applicable advanced options and save your datastore.

Task overview: HP Vertica [page 15]

Related Information

Enable MIT Kerberos for HP Vertica SSL protocol [page 16]
Creating a DSN for HP Vertica with Kerberos SSL [page 19]
Increasing loading speed for HP Vertica [page 23]
HP Vertica data type conversion [page 23]
HP Vertica table source [page 25]
HP Vertica target table configuration [page 26]
HP Vertica datastore options [page 22]

3.8.3.1 HP Vertica datastore options

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

The following tables contain datastore configuration options specific to HP Vertica datastore.

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	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. You must first install and configure MIT Kerberos 5 and perform other HP Vertica set up tasks before you can define a DSN. For more information about HP Vertica MIT Kerberos and DSN for HP Vertica, read the Server Management section of the *Administrator Guide*.
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*.

For more information about common datastore options and working with Aliases, see the Datastores section in the *Designer Guide*.

3.8.4 Increasing loading speed for HP Vertica

SAP Data Services does not support bulk loading for HP Vertica, but there are settings you can make to increase loading speed.

For complete details about connecting to HP Vertica, consult with the *Connecting to HP Vertica* guide at https://my.vertica.com/ (copy and paste URL in your browser to follow link). Select **Documentation** and click the applicable version from the dropdown list.

When you load data to an HP Vertica target in a data flow, the software automatically executes an HP Vertica statement that contains a `COPY Local` statement. This statement makes the ODBC driver read and stream the data file from the client to the server.

You can further increase loading speed by increasing rows per commit and enable use native connection load balancing:

1. when you configure the ODBC driver for HP Vertica, enable the option to use native connection load balancing.
2. In Designer, open the applicable data flow.
3. In the workspace, double-click the HP Vertica datastore target object to open it.
4. Open the *Options* tab in the lower pane.
5. Increase the number of rows in the *Rows per commit* option.

Task overview: HP Vertica [page 15]

Related Information

Enable MIT Kerberos for HP Vertica SSL protocol [page 16]
Creating a DSN for HP Vertica with Kerberos SSL [page 19]
Creating HP Vertica datastore with SSL encryption [page 21]
HP Vertica data type conversion [page 23]
HP Vertica table source [page 25]
HP Vertica target table configuration [page 26]

3.8.5 HP Vertica data type conversion

SAP Data Services converts incoming HP Vertica data types to native data types, and outgoing native data types to HP Vertica data types.

The following table contains HP Vertica data types and the native data types to which Data Services converts them.

HP Vertica data type	Data Services data type
Boolean	Int
Integer, INT, BIGINT, INT8, SMALLINT, TINYINT	Decimal
FLOAT	Double
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

The following table contains native data types and the HP Vertica data types to which Data Services outputs them. Data Services outputs the converted data types to HP Vertica template tables or Data_Transfer transform tables.

Data Services data type	HP Vertica data type
Blob	Long Varbinary
Date	Date
Datetime	Timestamp
Decimal	Decimal

Data Services data type	HP Vertica data type
Double	Float
Int	Int
Interval	Float
Long	Long Varchar
Real	Float
Time	Time
Varchar	Varchar
Timestamp	Timestamp

Parent topic: HP Vertica [page 15]

Related Information

Enable MIT Kerberos for HP Vertica SSL protocol [page 16]
Creating a DSN for HP Vertica with Kerberos SSL [page 19]
Creating HP Vertica datastore with SSL encryption [page 21]
Increasing loading speed for HP Vertica [page 23]
HP Vertica table source [page 25]
HP Vertica target table configuration [page 26]

3.8.6 HP Vertica table source

Configure options for an HP Vertica table as a source by opening the source editor in the data flow.

HP Vertica source table options

Option	Description
Table name	Specifies the table name for the source table.
Table owner	Specifies the table owner. You cannot edit the value. Data Services automatically populates with the name that you entered when you created the HP Vertica table.

Option	Description
Datastore name	Specifies the name of the related HP Vertica datastore.
Database type	Specifies the database type. You cannot edit this value. Data Services automatically populates with the database type that you chose when you created the datastore.

Parent topic: HP Vertica [page 15]

Related Information

Enable MIT Kerberos for HP Vertica SSL protocol [page 16]
Creating a DSN for HP Vertica with Kerberos SSL [page 19]
Creating HP Vertica datastore with SSL encryption [page 21]
Increasing loading speed for HP Vertica [page 23]
HP Vertica data type conversion [page 23]
HP Vertica target table configuration [page 26]

3.8.7 HP Vertica target table configuration

Configure options for an HP Vertica table as a target by opening the target editor in the data flow.

Options tab General options

Option	Description
Column comparison	Specifies how the software maps input columns to output columns: • *Compare by position*: Maps source columns to target columns by position, and ignores column names. • *Compare by name*: Maps source columns to target columns by column name. *Compare by name* is the default setting. Data Services issues validation errors when the data types of the columns do not match.

Option	Description
Number of loaders	Specifies the number of loaders Data Services uses to load data to the target. Enter a positive integer. The default is **1**. There are different types of loading: • Single loader loading: Loading with one loader. • Parallel loading: Loading with two or more loaders. With parallel loading, each loader receives the number of rows indicated in the *Rows per commit* option, and processes the rows in parallel with other loaders. **Example** For example, if *Rows per commit* = 1000 and *Number of Loaders* = 3: • First 1000 rows go to the first loader • Second 1000 rows go to the second loader • Third 1000 rows go to the third loader • Fourth 1000 rows go to the first loader

Options tab Error handling options

Option	Description
Use overflow file	Specifies whether Data Services uses a recovery file for rows that it could not load. • *No*: Data Services does not save information about unloaded rows. The default setting is *No*. • *Yes*: Data Services loads data to an overflow file when it cannot load a row. When you select *Yes*, also complete *File Name* and *File Format*.
File name *File format*	Specifies the file name and file format for the overflow file. Applicable only when you select *Yes* for *Use overflow file*. Enter a file name or specify a variable The overflow file can include the data rejected and the operation being performed (write_data) or the SQL command used to produce the rejected operation (write_sql).

Update control

Option	Description
Use input keys	Specifies whether Data Services uses the primary keys from the input table when the target table does not have a primary key. • *Yes*: Uses the primary keys from the input table when the target table does not have primary keys. • *No*: Does not use primary keys from the input table when the target table does not have primary keys. *No* is the default setting.
Update key columns	Specifies whether Data Services updates key column values when it loads data to the target table. • *Yes*: Updates key column values when it loads data to the target table. • *No*: Does not update key column values when it loads data to the target table. *No* is the default setting.
Auto correct load	Specifies whether Data Services uses auto correct loading when it loads data to the target table. Auto correct loading ensures that Data Services does not duplicate the same row in a target table. Auto correct load is useful for data recovery operations. • *Yes*: Uses auto correct loading. **i Note** Not applicable for targets in real time jobs or target tables that contain LONG columns. • *No*: Does not use auto correct loading. *No* is the default setting. For more information about auto correct loading, read about recovery mechanisms in the *Designer Guide*.
Ignore columns with value	Specifies a value that might appear in a source column and that you do not want updated in the target table during auto correct loading. Enter a string excluding single or double quotation marks. The string can include spaces. When Data Services finds the string in the source column, it does not update the corresponding target column during auto correct loading.

Transaction control

Option	Description
Include in transaction	Specifies that this target table is included in the transaction processed by a batch or real-time job. • *No*: This target table is not included in the transaction processed by a batch or real-time job. *No* is the default setting • *Yes*: The target table is included in the transaction processed by a batch or real-time job. Selecting *Yes* enables Data Services to commit data to multiple tables as part of the same transaction. If loading fails for any of the tables, Data Services does not commit any data to any of the tables. **i Note** Ensure that the tables are from the same datastore. Data Services does not push down a complete operation to the database when transactional loading is enabled. Data Services may buffer rows to ensure the correct load order. If the buffered data is larger than the virtual memory available, Data Services issues a memory error. If you choose to enable transactional loading, the following options are not available: • *Rows per commit* • *Use overflow file* and *overflow file specification* • *Number of loaders*

Parent topic: HP Vertica [page 15]

Related Information

Enable MIT Kerberos for HP Vertica SSL protocol [page 16]
Creating a DSN for HP Vertica with Kerberos SSL [page 19]
Creating HP Vertica datastore with SSL encryption [page 21]
Increasing loading speed for HP Vertica [page 23]
HP Vertica data type conversion [page 23]
HP Vertica table source [page 25]

3.9 MongoDB

The MongoDB adapter allows you to read data from MongoDB sources and load data to other SAP Data Services targets.

MongoDB is an open-source document database, which has JSON-like documents called BSON. MongoDB has dynamic schemas instead of traditional schema-based data.

Data Services needs metadata to gain access to MongoDB data for task design and execution. Use Data Services processes to generate schemas by converting each row of the BSON file into XML and converting XML to XSD.

Data Services uses the converted metadata in XSD files to access MongoDB data.

3.9.1 MongoDB metadata

Use data from MongoDB as a source or target in a data flow, and also create templates.

The embedded documents and arrays in MongoDB are represented as nested data. SAP Data Services converts MongoDB BSON files to XML and then to XSD. Data Services saves the XSD file to the following location: `LINK_DIR\ext\mongo\mcache`.

Restrictions and limitations

Data Services has the following restrictions and limitations for working with MongoDB:

- In the MongoDB collection, the tag name cannot contain special characters that are invalid for the XSD file. For example, the following special characters are invalid for XSD files: >, <, &,/, \, #, and so on. If special characters exist, Data Services removes them.
- MongDB data is always changing, so the XSD may not reflect the entire data structure of all the documents in the MongoDB.
- Data Services does not support projection queries on adapters.
- Data Services ignores any new fields that you add after the metadata schema creation that were not present in the common documents.
- Data Services does not support push down operators when you use MongoDB as a target.

For more information about formatting XML documents, see the Nested Data section in the *Designer Guide*. For more information about source and target objects, see the Data flows section of the *Designer Guide*.

Parent topic: MongoDB [page 30]

Related Information

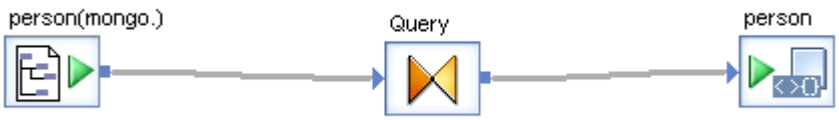
MongoDB as a source [page 31]
MongoDB as a target [page 34]
MongoDB template documents [page 36]
Preview MongoDB document data [page 38]
Parallel Scan [page 39]
Reimport schemas [page 40]
Searching for MongoDB documents in the repository [page 41]

3.9.2 MongoDB as a source

Use MongoDB as a source in Data Services and flatten the nested schema by using the XML_Map transform.

The following examples illustrate how to use various objects to process MongoDB sources in data flows.

Example 1: Change the schema of a MongoDB source using the Query transform, and load output to an XML target.



> **Note**
>
> Specify conditions in the Query transform. Some conditions can be pushed down and others are processed by Data Services.

Example 2: Set a dataflow where Data Services reads the schema and then loads the schema directly into an XML template file.

Example 3: Flatten a schema using the XML_Map transform and then load the data to a table or flat file.

> **Note**
>
> Specify conditions in the XML_Map transform. Some conditions can be pushed down and others are processed by Data Services.

Parent topic: MongoDB [page 30]

Related Information

MongoDB metadata [page 30]
MongoDB as a target [page 34]
MongoDB template documents [page 36]
Preview MongoDB document data [page 38]
Parallel Scan [page 39]
Reimport schemas [page 40]
Searching for MongoDB documents in the repository [page 41]
MongoDB query conditions [page 33]
Push down operator information [page 33]

3.9.2.1 MongoDB query conditions

Use query criteria to retrieve documents from a MongoDB collection.

Use query criteria as a parameter of the `db.<collection>.find()` method. Add MongoDB query conditions to a MongoDB table as a source in a data flow.

To add a MongoDB query format, enter a value next to the *Query criteria* parameter in the source editor *Adapter Source* tab. Ensure that the query criteria is in MongoDB query format. For example, `{ type: { $in: ['food', 'snacks'] } }`

> **Example**
>
> Given a value of `{prize:100}`, MongoDB returns only rows that have a field named "prize" with a value of 100. If you don't specify the value 100, MongoDB returns all the rows.

Configure a `Where` condition so that Data Services pushes down the condition to MongoDB. Specify a `Where` condition in a Query or XML_Map transform, and place the Query or XML_Map transform after the MongoDB source object in the data flow. MongoDB returns only the rows that you want.

For more information about the MongoDB query format, consult the MongoDB Web site.

> **Note**
>
> If you use the XML_Map transform, it may have a query condition with a SQL format. Data Services converts the SQL format to the MongoDB query format and uses the MongoDB specification to push down operations to the source database. In addition, be aware that Data Services does not support push down of query for nested arrays.

Related Information

Push down operator information [page 33]

3.9.2.2 Push down operator information

SAP Data Services processes push down operators with a MongoDB source in specific ways based on the circumstance.

Push down behavior:

- Data Services does not push down `Sort by` conditions.
- Data Services pushes down `Where` conditions.
- Data Services does not push down the nested array when you use a nested array in a `Where` condition.
- Data Services does not support push down operators when you use MongoDB as a target.

Data Services supports the following operators when you use MongoDB as a source:

- Comparison operators: `=`, `!=`, `>`, `>=`, `<`, `<=`, `like`, and `in`.
- Logical operators: `and` and `or` in SQL query.

3.9.3 MongoDB as a target

Configure options for MongoDB as a target in your data flow using the target editor.

About the `<_id>` field

SAP Data Services considers the `<_id>` field in MongoDB data as the primary key. If you create a new MongoDB document and include a field named `<_id>`, Data Services recognizes that field as the unique BSON ObjectID. If a MongoDB document contains more than one `<_id>` field at different levels, Data Services considers only the `<_id>` field at the first level as the BSON Object Id.

The following table contains descriptions for options in the *Adapter Target* tab of the target editor.

Adapter Target tab options

Option	Description
Use auto correct	Specifies the mode Data Services uses for MongoDB as a target datastore. • *True*: Uses Upsert mode for the writing behavior. Updates the document with the same `<_id>` field or it inserts a new `<_id>` field. **Note** Selecting *True* may slow the performance of writing operations. • *False*: Uses Insert mode for writing behavior. If documents have the same `<_id>` field in the MongoDB collection, then Data Services issues an error message.
Write concern level	Specifies the MongoDB write concern level that Data Services uses for reporting the success of a write operation. Enable or disable different levels of acknowledgement for writing operations. • *Acknowledged*: Provides acknowledgment of write operations on a standalone `mongod` or the primary in a replica set. *Acknowledged* is the default setting. • *Unacknowledged*: Disables the basic acknowledgment and only returns errors of socket exceptions and networking errors. • *Replica Set Acknowledged*: Guarantees that write operations have propagated successfully to the specified number of replica set members, including the primary. • *Journaled*: Acknowledges the write operation only after MongoDB has committed the data to a journal. • *Majority*: Confirms that the write operations have propagated to the majority of voting nodes.

Option	Description
Use bulk	Specifies whether Data Services executes writing operations in bulk. Bulk may provide better performance. • *True*: Runs write operation in bulk for a single collection to optimize the CRUD efficiency. If the write operation in a bulk is more than 1000, MongoDB automatically splits into multiple bulk groups. • *False*: Does not run write operation in bulk. For more information about bulk, ordered bulk, and bulk maximum rejects, see the MongoDB documentation at http://help.sap.com/disclaimer?site=http://docs.mongodb.org/manual/core/bulk-write-operations/.
Use ordered bulk	Specifies the order in which Data Services executes write operations: Serial or Parallel. • *True*: Executes write operations in serial. • *False*: Executes write operations in parallel. *False* is the default setting. MongoDB processes the remaining write operations even when there are errors.
Documents per commit	Specifies the maximum number of documents that are loaded to a target before the software saves the data. • Blank: Uses the maximum of 1000 documents. Blank is the default setting. • Enter any integer to specify a number other than 1000.
Bulk maximum rejects	Specifies the maximum number of acceptable errors before Data Services fails the job. **i Note** Data Services continues to load to the target MongoDB even when the job fails. Enter an integer. Enter `-1` so that Data Services ignores and does not log bulk loading errors. If the number of actual errors is less than, or equal to the number you specify here, Data Services allows the job to succeed and logs a summary of errors in the adapter instance trace log. Applicable only when you select *True* for *Use ordered bulk*.
Delete data before loading	Deletes existing documents in the current collection before loading occurs. Retains all the configuration, including indexes, validation rules, and so on.
Drop and re-create	Specifies whether Data Services drops the existing MongoDB collection and creates a new one with the same name before loading occurs. • *True*: Drops the existing MongoDB collection and creates a new one with the same name before loading. Ignores the value of *Delete data before loading*. *True* is the default setting. • *False*: Does not drop the existing MongoDB collection and create a new one with the same name before loading. This option is available for template documents only.

Option	Description
Use audit	Specifies whether Data Services creates audit files that contain write operation information. • *True*: Creates audit files that contain write operation information. Stores audit files in the `<DS_COMMON_DIR>/adapters/audits/` directory. The name of the file is `<MongoAdapter_instance_name>.txt`. • *False*: Does not create and store audit files. Data Services behaves in the following way when a regular load fails: • *Use audit* = *False*: Data Services logs loading errors in the job trace log. • *Use audit* = *True*: Data Services logs loading errors in the job trace log and in the audit log. Data Services behaves in the following way when a bulk load fails: • *Use audit* = *False*: Data Services creates a job trace log that provides only a summary. It does not contain details about each row of bad data. There is no way to obtain details about bad data. • *Use audit* = *True*: Data Services creates a job trace log that provides only a summary but no details. However, the job trace log provides information about where to find details about each row of bad data in the audit file.

Parent topic: MongoDB [page 30]

Related Information

MongoDB metadata [page 30]
MongoDB as a source [page 31]
MongoDB template documents [page 36]
Preview MongoDB document data [page 38]
Parallel Scan [page 39]
Reimport schemas [page 40]
Searching for MongoDB documents in the repository [page 41]

3.9.4 MongoDB template documents

Use template documents as a target in one data flow or as a source in multiple data flows.

Template documents are useful in early application development when you design and test a project. After you import data for the MongoDB datastore, Data Services stores the template documents in the object library. Find template documents in the *Datastore* tab of the object library.

When you import a template document, the software converts it to a regular document. You can use the regular document as a target or source in your data flow.

> **i Note**
>
> Template documents are available in Data Services 4.2.7 and later. If you upgrade from a previous version, open an existing MongoDB datastore and then click *OK* to close it. Data Services updates the datastore so that you see the Template Documents node and any other template document related options.

Template documents are similar to template tables. For information about template tables, see the Data Services *User Guide* and the *Reference Guide*.

Parent topic: MongoDB [page 30]

Related Information

MongoDB metadata [page 30]
MongoDB as a source [page 31]
MongoDB as a target [page 34]
Preview MongoDB document data [page 38]
Parallel Scan [page 39]
Reimport schemas [page 40]
Searching for MongoDB documents in the repository [page 41]

3.9.4.1 Creating MongoDB template documents

Create MongoDB template documents as targets in data flows, then use the target as a source in a different data flow.

To use a MongoDB template as the target or source in a data flow, first use the template as a target. To add a MongoDB template as a target in a data flow, perform the following steps to create the target:

1. Click the template icon from the tool palette.
2. Click inside a data flow in the workspace.

   The *Create Template* dialog box opens.
3. Enter a name for the template in *Template name*.

   > **i Note**
   >
   > Use the MongoDB collection namespace format: `database.collection`. Don't exceed 120 bytes.
4. Select the related MongoDB datastore from the *In datastore* dropdown list.
5. Click *OK*.
6. To use the template document as a target in the data flow, connect the template document to the object that comes before the template document.

   Data Services automatically generates a schema based on the object directly before the template document in the data flow.

> ! Restriction
>
> The field `<_id>` is the default primary key of the MongoDB collection. Therefore, make sure that you correctly configure the `<_id>` field in the output schema of the object that comes directly before the target template. If you don't include `<_id>` in the output schema, the following error appears when you view the data: "`An element named <_id> present in the XML data input does not exist in the XML format used to set up this XML source in the data flow <dataflow>. Validate your XML data.`"

7. Click *Save*.

   The template document icon in the data flow changes, and Data Services adds the template document to the object library. Find the template document in the applicable database node under Templates.

Convert the template document into a regular document by selecting to import the template document in the object library. Then you can use the template document as a source or a target document in other data flows.

3.9.4.2 Convert a template document into a regular document

SAP Data Services enables you to convert an imported template document into a regular document.

Use one of the following methods to import a MongoDB template document:

- Open a data flow and select one or more template target documents in the workspace. Right-click, and choose *Import Document*.
- Select one or more template documents in the Local Object Library, right-click, and choose *Import Document*.

The icon changes and the document appears under *Documents* instead of *Template Documents* in the object library.

> i Note
>
> The *Drop and re-create* target configuration option is available only for template target documents. Therefore it is not available after you convert the template target into a regular document.

3.9.5 Preview MongoDB document data

Use the data preview feature in SAP Data Services Designer to view a sampling of data from a MongoDB document.

Choose one of the following methods to preview MongoDB document data:

- Expand an applicable MongoDB datastore in the object library. Right-click the MongoDB document and select *View Data* from the dropdown menu.
- Right-click the MongoDB document in a data flow and select *View Data* from the dropdown menu.

- Click the magnifying glass icon in the lower corner of either a MongoDB source or target object in a data flow.

  > **Note**
  >
  > By default, Data Services displays a maximum of 100 rows. Change this number by setting the *Rows To Scan* option in the applicable MongoDB datastore editor. Entering `-1` displays all rows.

For more information about viewing data, see the *Designer Guide*.

Parent topic: MongoDB [page 30]

Related Information

MongoDB metadata [page 30]
MongoDB as a source [page 31]
MongoDB as a target [page 34]
MongoDB template documents [page 36]
Parallel Scan [page 39]
Reimport schemas [page 40]
Searching for MongoDB documents in the repository [page 41]
MongoDB adapter datastore configuration options

3.9.6 Parallel Scan

SAP Data Services uses the MongoDB Parallel Scan process to improve performance while it generates metadata for big data.

To generate metadata, Data Services first scans all documents in the MongoDB collection. This scanning can be time consuming. However, when Data Services uses the Parallel Scan command `parallelCollectionScan`, it uses multiple parallel cursors to read all the documents in a collection. Parallel Scan can increase performance.

> **Note**
>
> Parallel Scan works with MongoDB server version 2.6.0 and above.

For more information about the `parallelCollectionScan` command, consult your MongoDB documentation.

For more information about Mongo adapter datastore configuration options, see the *Supplement for Adapters*.

Parent topic: MongoDB [page 30]

Related Information

MongoDB metadata [page 30]
MongoDB as a source [page 31]
MongoDB as a target [page 34]
MongoDB template documents [page 36]
Preview MongoDB document data [page 38]
Reimport schemas [page 40]
Searching for MongoDB documents in the repository [page 41]

3.9.7 Reimport schemas

When you reimport documents from your MongoDB datastore, SAP Data Services uses the current datastore settings.

Reimport a single MongoDB document by right-clicking the document and selecting *Reimport* from the dropdown menu.

To reimport all documents, right-click an applicable MongoDB datastore or right-click on the *Documents* node and select *Reimport All* from the dropdown menu.

> **i** Note
>
> When you enable *Use Cache*, Data Services uses the cached schema.
>
> When you disable *Use Cache*, Data Services looks in the sample directory for a sample BSON file with the same name. If there is a matching file, the software uses the schema from the BSON file. If there isn't a matching BSON file in the sample directory, the software reimports the schema from the database.

Parent topic: MongoDB [page 30]

Related Information

MongoDB metadata [page 30]
MongoDB as a source [page 31]
MongoDB as a target [page 34]
MongoDB template documents [page 36]
Preview MongoDB document data [page 38]
Parallel Scan [page 39]
Searching for MongoDB documents in the repository [page 41]

3.9.8 Searching for MongoDB documents in the repository

SAP Data Services enables you to search for MongoDB documents in your repository from the object library.

1. Right-click in any tab in the object library and choose *Search* from the dropdown menu.

   The *Search* dialog box opens.
2. Select the applicable MongoDB datastore name from the *Look in* dropdown menu.

   The datastore is the one that contains the document for which you are searching.
3. Select *Local Repository* to search the entire repository.
4. Select *Documents* from the *Object Type* dropdown menu.
5. Enter the criteria for the search.
6. Click *Search*.

   Data Services lists matching documents in the lower pane of the *Search* dialog box. A status line at the bottom of the *Search* dialog box shows statistics such as total number of items found, amount of time to search, and so on.

For more information about searching for objects, see the Objects section of the *Designer Guide*.

Task overview: MongoDB [page 30]

Related Information

MongoDB metadata [page 30]
MongoDB as a source [page 31]
MongoDB as a target [page 34]
MongoDB template documents [page 36]
Preview MongoDB document data [page 38]
Parallel Scan [page 39]
Reimport schemas [page 40]

3.10 Apache Impala

Create an ODBC datastore to connect to Apache Impala in Hadoop.

Before you create an Apache Impala datastore, download the Cloudera ODBC driver and create a data source name (DSN). Use the datastore to connect to Hadoop and import Impala metadata. Use the metadata as a source or target in a data flow.

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

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

For more information about ODBC datastores, see the Datastores section in the *Designer Guide*.

For descriptions of common datastore options, see the *Designer Guide*.

Download the Cloudera ODBC driver for Impala [page 42]
For Linux users. Before you create an Impala database datastore, connect to Apache Impala using the Cloudera OBDC driver.

Creating an Apache Impala datastore [page 43]
To connect to your Hadoop files and access Impala data, create an ODBC datastore in SAP Data Services Designer.

3.10.1 Download the Cloudera ODBC driver for Impala

For Linux users. Before you create an Impala database datastore, connect to Apache Impala using the Cloudera OBDC driver.

Perform the following high-level steps to download a Cloudera ODBC driver and create a data source name (DSN). 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):

   Select the driver that is compatible with your platform. For information about the correct driver versions, see the SAP Product Availability Matrix (PAM).
3. Start DSConnectionManager.sh.

   Either open the file or run the following command:

```
cd $LINK_DIR/bin/
$ ./DSConnectionManager.sh
```

Example

The following shows prompts and values in DS Connection Manager that includes Kerberos and SSL:

```
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:
<New DSN file name>
Specify the User Name:
<Hadoop user name>
Type database password:(no echo)
  *Type the Hadoop password. Password does not appear after you type it for security.
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:
  /<path>/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.
```

Task overview: Apache Impala [page 41]

Related Information

Creating an Apache Impala datastore [page 43]

3.10.2 Creating an Apache Impala datastore

To connect to your Hadoop files and access Impala data, create an ODBC datastore in SAP Data Services Designer.

Before performing the following steps, enable Impala Services on your Hadoop server. Then download the Cloudera driver for your platform.

> **i** Note
>
> If you didn't create a DSN (data source name) in Windows ODBC Data Source application, you can create a DSN in the following process.

To create an ODBC datastore for Apache Impala, perform the following steps in Designer:

1. Select *Tools* > *New* > *Datastore.*

   The datastore editor opens.
2. Choose *Database* from the *Datastore Type* dropdown list.
3. Choose *ODBC* from the *Database Type* dropdown list.
4. Select *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.
6. Select **Configure**.

The **Cloudera ODBC Driver for Impala DSN Setup** dialog box opens.

7. Enter the required information based on your system
8. Select **Advanced Options**.
9. Check **Use SQL Unicode Types**.
10. Close **Advanced Options**.
11. **Optional:** Enable Kerberos authentication by performing the following substeps:
    a. Choose **Kerberos** from the **Mechanism** list.
    b. Enter the name of the applicable realm in **Realm**.

       A realm is a set of managed nodes that share the same Kerberos database.
    c. Enter the fully qualified domain name (FQDN) of the Hive Server host in **Host FQDN**.
    d. Enter the service principal name for the Hive server in **Service Name**.
    e. Enable the *Canonicalize Principal FQDN* option.

       This option canonicalizes the host FQDN in the server principal name.
12. **Optional:** To enable Secure Sockets Layer (SSL) protocol, perform the following substeps:
    a. Choose **No Authentication (SSL)** from the **Mechanism** list.
    b. Select **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.
    d. Close **Advanced Options**.
13. Close the **Cloudera ODBC Driver for Impala DSN Setup** dialog box and the ODBC Data Source Administrator.
14. Select the Cloudera DSN that you created from the *Data Source Name* list.

    The DSN appears in the dropdown list only after you've created it.
15. Select *Advanced* and complete the advanced options as necessary.
    a. **Optional:** Set the *Code page* option to *utf-8* in the **Locale** group to process multibyte data in Impala tables.
    b. **Optional:** In the *ODBC Date Function Support* group, set the *Week* option to *No*.

       If you don't set the *Week* option to *No*, the result of the Data Services built-in function `week_in_year()` may be incorrect.

Task overview: Apache Impala [page 41]

Related Information

Download the Cloudera ODBC driver for Impala [page 42]

3.11 PostgreSQL

To use your PostgreSQL tables as sources and targets in SAP Data Services, create a PostgreSQL datastore and import your tables and other metadata.

Required versions

Download and install the latest supported PostgreSQL Server version from the official PostgreSQL Web site. Check the Product Availability Matrix (PAM) on the SAP Support Portal to ensure that you have the supported PostgreSQL version for your version of Data Services.

> **Note**
>
> Data Services supports PostgreSQL on Windows beginning with version 4.2.12 (14.02.12.00). Data Services supports PostgreSQL on Linux beginning with version 4.2.12 Patch 1 (14.02.12.01).

Obtain the ODBC driver that is compatible with your version of PostgreSQL. To avoid potential processing problems, download the ODBC driver from the official PostgreSQL Web site.

DSN or DSN-less connections

Create a PostgreSQL datastore using either a DSN or DSN-less connection.

> **Note**
>
> Currently, Data Services does not support SSL connections for PostgreSQL.

Bulk loading

Configure the PostgreSQL datastore for bulk loading to the target PostgreSQL database. Before you can configure the bulk loading options, obtain the PSQL tool from the official PostgreSQL website. In addition to setting the bulk loader directory and the location for the PSQL tool in the datastore, complete the *Bulk Loader Options* tab in the target editor. For complete information about the bulk loading in PostgreSQL and the options, see the *Performance Optimization Guide*.

> **Note**
>
> Data Services supports bulk loading for PostgreSQL DSN-less connections only.

Pushdown functions

Data Services supports the basic pushdown functions for PostgreSQL. For a list of pushdown functions that Data Services supports for PostgreSQL, see 2212730.

UTF-8 encoding

To process PostgreSQL tables as sources in data flows, Data Services requires that all data in PostgreSQL tables use UTF-8 encoding. Additionally, Data Services outputs data to PostgreSQL target tables using UTF 8 encoding.

Conversion to or from internal data types

Data Services converts PostgreSQL data types to data types that it can process. After processing, Data Services outputs data and converts the data types back to the corresponding PostgreSQL data types.

Datastore options for PostgreSQL [page 46]
Complete options in the datastore editor to set the datastore type, database version, database access information, and DSN information if applicable.

Configure the PostgreSQL ODBC driver [page 49]
Configure the PostgreSQL ODBC driver for Windows or Linux to update the configuration file with the applicable driver information.

Import PostgreSQL metadata [page 49]
Use the PostgreSQL database datastore to access the schemas and tables in the defined database.

PostgreSQL source, target, and template tables [page 50]
Use PostgreSQL tables as sources and targets in data flows and use PostgreSQL table schemas for template tables.

PostgreSQL data type conversions [page 50]
When you import metadata from a PostgreSQL table into the repository, SAP Data Services converts PostgreSQL data types to Data Services native data types for processing.

3.11.1 Datastore options for PostgreSQL

Complete options in the datastore editor to set the datastore type, database version, database access information, and DSN information if applicable.

The first set of options define the datastore type (database) and the PostgreSQL version information.

PostgreSQL datastore option descriptions

Option	Value
Datastore Type	Select *Database*.
Database Type	Select *PostgreSQL*.
Database Version	Select a version. Check the Product Availability Matrix (PAM) to ensure that you have the supported PostgreSQL version for your version of Data Services.

To create a server-name (DSN-less) datastore for PostgreSQL, complete the database-specific options described in the following table.

PostgreSQL database option descriptions for DSN-less connection

Option	Description
Database server name	Specifies the database server address. Enter **`localhost`** or an IP address.
Database name	Specifies the database name to which this datastore connects.
Port	Specifies the port number that this datastore uses to access the database.
User name	Specifies the name of the user authorized to access the database.
Password	Specifies the password related to the specified *User name*.
Enable Automatic Data Transfer	Specifies that any data flow that uses the tables imported with this datastore may use the Data_Transfer transform. Data_Transfer uses transfer tables to push down certain operations to the database server for more efficient processing.
Bulk loader directory	Specifies the directory where Data Services stores the files related to bulk loading, such as log file, error file, and temporary files. Click the down arrow at the end of the field and select `<Browse>` or select an existing global variable that you created for this location. If you leave this field blank, Data Services writes the files to `%DS_COMMON_DIR%/log/bulkloader`. **Note** Bulk loading in PostgreSQL is applicable for DSN-less connections only.

Option	Description
PSQL full path	Specifies the full path to the location of the PSQL tool. The PSQL tool is required for PostgreSQL bulk loading. Obtain the PSQL tool from the official PostgreSQL website. For convenience, create a global variable for this value before you configure the datastore. Click the down arrow at the end of the field and select `<Browse>` or select the global variable. **i Note** Bulk loading in PostgreSQL is applicable for DSN-less connections only.

When you create a DSN connection, the remaining options include information that you don't enter in the ODBC Data Source Administrator.

PostgreSQL DSN option descriptions

Option	Description
Data Source Name	Specifies the name of the DSN you create in the *ODBC Data Source Administrator*. Ensure that you create the DSN so that it appears in the dropdown list.
User Name	Specifies the user name to access the data source defined in the DSN.
Password	Specifies the password related to the User Name value.

For a list of properties required for each database type, see the *Administrator Guide*.

For information about bulk loading with PostgreSQL, see the *Performance Optimization Guide*.

Parent topic: PostgreSQL [page 45]

Related Information

Configure the PostgreSQL ODBC driver [page 49]
Import PostgreSQL metadata [page 49]
PostgreSQL source, target, and template tables [page 50]
PostgreSQL data type conversions [page 50]

3.11.2 Configure the PostgreSQL ODBC driver

Configure the PostgreSQL ODBC driver for Windows or Linux to update the configuration file with the applicable driver information.

For Windows, use the ODBC Drivers Selector to verify the ODBC driver is installed. For Linux, configure the ODBC driver using the SAP Data Services Connection Manager. For information about configuring the ODBC driver, see the Server Maintenance section of the *Administrator Guide*.

Parent topic: PostgreSQL [page 45]

Related Information

Datastore options for PostgreSQL [page 46]
Import PostgreSQL metadata [page 49]
PostgreSQL source, target, and template tables [page 50]
PostgreSQL data type conversions [page 50]

3.11.3 Import PostgreSQL metadata

Use the PostgreSQL database datastore to access the schemas and tables in the defined database.

Open the datastore and view the metadata available to download. For PostgreSQL, download schemas and the related tables. Each table resides under a specific schema. For example, each schema contains tables that use the schema. A table name appears as `<dbname>.<schema_name>.<table_name>`.

Import metadata by browsing, by name, or by searching.

For more information about viewing metadata, see the Datastore metadata section of the *Designer Guide*.

For more information about the imported metadata from database datastores, see the Datastores section of the *Designer Guide*.

Parent topic: PostgreSQL [page 45]

Related Information

Datastore options for PostgreSQL [page 46]
Configure the PostgreSQL ODBC driver [page 49]
PostgreSQL source, target, and template tables [page 50]
PostgreSQL data type conversions [page 50]

3.11.4 PostgreSQL source, target, and template tables

Use PostgreSQL tables as sources and targets in data flows and use PostgreSQL table schemas for template tables.

Drag the applicable PostgreSQL table onto your workspace and connect it to a data flow as a source or target. Also, use a template table as a target in a data flow and save it to use as a future source in a different data flow.

See the *Designer Guide* to learn about using template tables. Additionally, see the Reference Guide for descriptions of options to complete for source, target, and template tables.

Parent topic: PostgreSQL [page 45]

Related Information

Datastore options for PostgreSQL [page 46]
Configure the PostgreSQL ODBC driver [page 49]
Import PostgreSQL metadata [page 49]
PostgreSQL data type conversions [page 50]

3.11.5 PostgreSQL data type conversions

When you import metadata from a PostgreSQL table into the repository, SAP Data Services converts PostgreSQL data types to Data Services native data types for processing.

After processing, Data Services converts data types back to PostgreSQL data types when it outputs the generated data to the target.

The following table contains PostgreSQL data types and the corresponding Data Services data types.

Data type conversion for PostgreSQL

PostgreSQL data type	Converts to or from Data Services data type	Notes
Boolean/Integer/Smallint	Int	
Serial/SamlIserial/Serial4/OID	Int	
Bigint/BigSerial/Serial8	Decimal(19,0)	
Float(1)-Float(24), Real	real	
Float(25)-Float(53), Double precision	double	
Money	double	
Numeric(precision, scale)	Decimal(precision, scale)	
Numeric/Decimal	Decimal(28,6)	

PostgreSQL data type	Converts to or from Data Services data type	Notes
Bytea	Blob	
Char(n)	Fixedchar(n)	
Text/varchar(n)	Varchar(n)	
DATE	Date	
TIMESTAMP	Datetime	
TIMESTAMPTZ	Varchar(127)	
TIMETZ	Varchar(127)	
INTERVAL	Varchar(127)	

If Data Services encounters a column that has an unsupported data type, it does not import the column. However, you can configure Data Services to import unsupported data types by checking the *Import unsupported data types as VARCHAR of size* option in the datastore editor dialog box.

> **i** Note
>
> When you import tables that have specific PostgreSQL native data types, Data Services saves the data type as varchar or integer, and includes an attribute setting for Native Type. The following table contains the column data type in which Data Services saves the PostgreSQL native data type, and the corresponding attribute.
>
> Data Services saves PostgreSQL native data types
>
PostgreSQL column native data type	Data Services saves as data type	Data Services attribute
json	Varchar	Native Type = JSON
jsonb		Native Type = JSONB
xml		Native Type = XML
uuid		Native Type = UUID
bool	Integer	Native Type = BOOL
text		Native Type = TEXT
bigint		Native Type = INT8

Parent topic: PostgreSQL [page 45]

Related Information

Datastore options for PostgreSQL [page 46]
Configure the PostgreSQL ODBC driver [page 49]
Import PostgreSQL metadata [page 49]
PostgreSQL source, target, and template tables [page 50]

3.12 SAP HANA

Process your SAP HANA data in SAP Data Services by creating an SAP HANA database datastore.

Import SAP HANA metadata using a database datastore. Use the table metadata as sources and targets in data flows. Some of the benefits of using SAP HANA in Data Services include the following:

- Protect your SAP HANA data during network transmission using SSL/TLS protocol and X.509 authentication with Cryptographic libraries.
- Create stored procedures and enable bulk loading for faster reading and loading.
- Load spatial and complex spatial data from Oracle to SAP HANA.

> **i Note**
>
> Beginning with SAP HANA 2.0 SP1, access databases only through a multitenant database container (MDC). If you use a version of SAP HANA that is earlier than 2.0 SP1, access only a single database.

3.12.1 Cryptographic libraries and global.ini settings

When you create an SAP HANA database datastore with SSL/TLS encryption or X.509 authentication, configure both server side and client side for the applicable authentication.

On the server side, the process of configuring the ODBC driver, SSL/TLS, and/or X.509 authentication automatically sets the applicable settings in the communications section of the `global.ini` file.

SAP HANA uses the SAP CommonCrypto library for both SSL/TLS encryption and X.509 authentication. The SAP HANA server installer installs the CommonCryptoLib (libsapcrypto.sar) to `$DIR_EXECUTABLE` by default.

> **i Note**
>
> SAP CommonCrypto library was formerly known as SAPCrypto library.

> **i Note**
>
> Support for OpenSSL in SAP HANA is deprecated. If you are using OpenSSL, we recommend that you migrate to CommonCryptoLib. For more information, see 2093286

Obtaining the SAP CommonCryptoLib file in Windows and Unix [page 53]
The SAP CommonCryptoLib files are required for using SSL/TLS encryption and X.509 authentication in your SAP HANA database datastores.

Parent topic: SAP HANA [page 52]

Related Information

X.509 authentication [page 54]
Bulk loading in SAP HANA [page 55]
Creating stored procedures in SAP HANA [page 58]
SAP HANA database datastores [page 59]
Configuring DSN for SAP HANA on Windows [page 65]
Configuring DSN for SAP HANA on Unix [page 66]
Datatype conversion for SAP HANA [page 68]
Using spatial data with SAP HANA [page 70]

3.12.1.1 Obtaining the SAP CommonCryptoLib file in Windows and Unix

The SAP CommonCryptoLib files are required for using SSL/TLS encryption and X.509 authentication in your SAP HANA database datastores.

If you use the SAP HANA ODBC driver version 2.9 or higher, you don't have to perform the following steps to obtain the SAP CommonCrypto library, because it's bundled with the driver.

To obtain the SAP CommonCryptoLib file, perform the following steps based on your platform:

1. For Windows:
   a. Create a local folder to store the CommonCryptoLib files.
   b. Download and install the applicable version of SAPCAR from the SAP download center.

      Use SAPCAR to extract the SAP CommonCryptoLib libraries.
   c. Obtain the SAP CommonCryptoLib Library file from the SAP download center.
   d. Use SAPCar to extract the library files from `libsapcrypto.sar` to the local folder that you created to store the files.
   e. Create a system variable in Windows named `$SECUDIR` and point to the local folder that you created for the CommonCryptoLib library files.

      The process to create a system variable on Windows varies based on your Windows version.

      > **Example**
      >
      > To create a system variable for Windows 10 Enterprise, access Control Panel as an administrator and open Systems. Search for System Variables and select Edit System Variables.
   f. Append `%SECUDIR%` to the `PATH` variable in Environment Variables.
   g. Restart Windows.
2. For Unix:
   a. Create a local folder to store the CommonCryptoLib files.
   b. Obtain the SAP CommonCryptoLib Library `libsapcrypto.sar` file from the Software Center.
   c. Use SAPCar to extract the library files from `libsapcrypto.sar` to the local folder that you created to store the files.
   d. Create a system variable named `SECUDIR` and point to the local folder that you created to store the files.

      ```
      export SECUDIR=/PATH/<LOCAL_FOLDER>
      ```
   e. Append `$SECUDIR` to `PATH`.

      ```
      export PATH=$SECUDIR:$PATH
      ```
   f. Restart the Job Server.

Task overview: Cryptographic libraries and global.ini settings [page 53]

3.12.2 X.509 authentication

X.509 authentication is a more secure method of accessing SAP HANA than user name and password authentication.

Include X.509 authentication when you configure an SAP HANA datastore. When you use X.509 authentication, consider the following information:

- Use with or without SSL/TLS protocol.
- Applicable for both DSN and server named (DSN-less) connections.

- Uses the same Cryptographic libraries and global.ini settings as the SSL/TLS protocol.
- Requires an X.509 key store file, which contains the following:
  - The X.509 client certificate.
  - The SSL server certificate, only if you select to use SSL/TLS encryption and you select to validate the server certificate in SAP Data Services.

> **i** Note
>
> Support for X.509 authentication begins in SAP Data Services version 4.2 SP14 (14.2.14.16) and is applicable for SAP HANA Server on premise 2.0 SP05 revision 56 and above, and SAP HANA Cloud. For SAP HANA client, X.509 is applicable for SAP HANA ODBC client 2.7 and above.

Create and configure the server and client certificate files before you include X.509 authentication for connecting to your SAP HANA data. For details, see the SAP Knowledge Base article 3126555.

Parent topic: SAP HANA [page 52]

Related Information

Cryptographic libraries and global.ini settings [page 53]
Bulk loading in SAP HANA [page 55]
Creating stored procedures in SAP HANA [page 58]
SAP HANA database datastores [page 59]
Configuring DSN for SAP HANA on Windows [page 65]
Configuring DSN for SAP HANA on Unix [page 66]
Datatype conversion for SAP HANA [page 68]
Using spatial data with SAP HANA [page 70]

3.12.3 Bulk loading in SAP HANA

SAP Data Services improves bulk loading for SAP HANA by using a staging mechanism to load data to the target table.

When Data Services uses changed data capture (CDC) or auto correct load, it uses a temporary staging table to load the target table. Data Services loads the data to the staging table and applies the operation codes INSERT, UPDATE, and DELETE to update the target table. With the *Bulk load* option selected in the target table editor, any one of the following conditions triggers the staging mechanism:

- The data flow contains a Map CDC Operation transform.
- The data flow contains a Map Operation transform that outputs UPDATE or DELETE rows.
- The data flow contains a Table Comparison transform.
- The *Auto correct load* option in the target table editor is set to *Yes*.

If none of these conditions are met, the input data contains only INSERT rows. Therefore Data Services performs only a bulk insert operation, which does not require a staging table or the need to execute any additional SQL.

By default, Data Services automatically detects the SAP HANA target table type. Then Data Services updates the table based on the table type for optimal performance.

The bulk loader for SAP HANA is scalable and supports UPDATE and DELETE operations. Therefore, the following options in the target table editor are also available for bulk loading:

- *Use input keys*: Uses the primary keys from the input table when the target table does not contain a primary key.
- *Auto correct load*: If a matching row to the source table does not exist in the target table, Data Services inserts the row in the target. If a matching row exists, Data Services updates the row based on other update settings in the target editor.

Find these options in the target editor under *Update Control*.

For more information about SAP HANA bulk loading and option descriptions, see the *Data Services Supplement for Big Data*.

SAP HANA target table options [page 56]
When you use SAP HANA tables as targets in a data flow, configure options in the target editor.

Parent topic: SAP HANA [page 52]

Related Information

Cryptographic libraries and global.ini settings [page 53]
X.509 authentication [page 54]
Creating stored procedures in SAP HANA [page 58]
SAP HANA database datastores [page 59]
Configuring DSN for SAP HANA on Windows [page 65]
Configuring DSN for SAP HANA on Unix [page 66]
Datatype conversion for SAP HANA [page 68]
Using spatial data with SAP HANA [page 70]
SAP HANA target table options [page 56]

3.12.3.1 SAP HANA target table options

When you use SAP HANA tables as targets in a data flow, configure options in the target editor.

The following tables describe options in the target editor that are applicable to SAP HANA. For descriptions of the common options, see the *Reference Guide*.

Options

Option	Description
Table type	Specifies the table type when you use SAP HANA template table as target. • *Column Store*: Creates tables organized by column. *Column Store* is the default setting. **i Note** Data Services does not support blob, dbblob, and clob data types for column store table types. • *Row Store*: Creates tables organized by row.

Bulk loading

Option	Description
Bulk load	Specifies whether Data Services uses bulk loading to load data to the target. • **Selected:** Uses bulk loading to load data to the target. • **Not selected:** Does not use bulk loading to load data to the target.
Mode	Specifies the mode that Data Services uses for loading data to the target table: • *Append*: Adds new records to the table. *Append* is the default setting. • *Truncate*: Deletes all existing records in the table and then adds new records.

Commit size	Specifies the maximum number of rows that Data Services loads to the staging and target tables before it saves the data (commits).
	• *default*: Uses a default commit size based on the target table type.
	  • **Column Store:** Default commit size is 10,000
	  • **Row Store:** Default commit size is 1,000
	• Enter a value that is greater than 1.

Update method	Specifies how Data Services applies the input rows to the target table.
	• *default*: Uses an update method based on the target table type:
	  • **Column Store:** Uses UPDATE to apply the input rows.
	  • **Row Store:** Uses DELETE-INSERT to apply the input rows.
	• *UPDATE*: Issues an UPDATE to the target table.
	• *DELETE-INSERT*: Issues a DELETE to the target table for data that matches the old data in the staging table. Issues an INSERT with the new data.
	i Note
	Do not use DELETE-INSERT if the update rows contain data for only some of the columns in the target table. If you use DELETE-INSERT, Data Services replaces missing data with NULLs.

Parent topic: Bulk loading in SAP HANA [page 55]

Related Information

3.12.4 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.

For more information about creating stored procedurees in a database, see the Functions and Procedures section in the *Designer Guide*.

Parent topic: SAP HANA [page 52]

Related Information

Cryptographic libraries and global.ini settings [page 53]

3.12.5 SAP HANA database datastores

To access SAP HANA data for SAP Data Services processes, configure an SAP HANA database datastore with either a data source name (DSN) or a server name (DSN-less) connection.

You can optionally include secure socket layer (SSL) or transport layer security (TLS) for secure transfer of data over a network, and you can use X.509 authentication instead of user name and password authentication.

> **i Note**
>
> Support for SSL/TLS with a DSN connection begins in SAP Data Services version 4.2 SP7 (14.2.7.0). Support for SSL/TLS with a DSN-less connection begins in SAP Data Services version 4.2 SP12 (14.2.12.0).

> **i Note**
>
> Support for X.509 authentication begins in SAP Data Services version 4.2 SP14 (14.2.14.16) and is applicable for SAP HANA Server on premise 2.0 SP05 revision 56 and above and SAP HANA Cloud. For SAP HANA client, X.509 is applicable for SAP HANA ODBC client 2.7 and above.

When you create an SAP HANA datastore, and use SAP HANA data in data flows, Data Services requires the SAP HANA ODBC driver. The following table lists the additional requirements for including additional authentications.

Authentication	Requirements
SSL/TLS	SAP CommonCrypto library SAP HANA SSL/TLS certificate and key files
X.509	SAP CommonCrypto library X.509 KeyStore file, which contains the following certificates: • X.509 client certificate. • SSL/TLS server certificate, only when you use SSL encryption and validate the server certificate in Data Services.

For more information about SAP HANA, SSL/TLS, SAP CommonCrypto library, and settings for secure external connections in the `global.ini` file, see the "SAP HANA Network and Communication Security" section of the SAP HANA Security Guide.

> **Note**
>
> Enabling SSL/TLS encryption slows job performance.

SAP HANA datastore prerequisites [page 60]
Before you configure an SAP HANA datastore, perform prerequisite tasks, such as configuring the ODBC driver.

SAP HANA datastore option descriptions [page 61]
In addition to the common database datastore options, SAP Data Services requires that you set options specific to SAP HANA, SSL/TLS, and X.509 authentication.

Parent topic: SAP HANA [page 52]

Related Information

Cryptographic libraries and global.ini settings [page 53]
X.509 authentication [page 54]
Bulk loading in SAP HANA [page 55]
Creating stored procedures in SAP HANA [page 58]
Configuring DSN for SAP HANA on Windows [page 65]
Configuring DSN for SAP HANA on Unix [page 66]
Datatype conversion for SAP HANA [page 68]
Using spatial data with SAP HANA [page 70]

3.12.5.1 SAP HANA datastore prerequisites

Before you configure an SAP HANA datastore, perform prerequisite tasks, such as configuring the ODBC driver.

Perform the following prerequisite tasks before you create the SAP HANA datastore:

- Download, install, and configure the SAP HANA ODBC driver.
  - For Windows, configure the driver using the ODBC Drivers Selector utility.
  - For UNIX, configure the driver using the ODBC Data Source Administrator.

  > **Note**
  >
  > If you plan to include SSL/TLS or X.509 authentication in the datastore, consider using the SAP HANA ODBC driver version 2.9 or higher. Version 2.9 or higher is bundled with the SAP CommonCrypto library, which is required for SSL/TLS and X.509.
- If you use a data source name (DSN), create a DSN connection using the ODBC Data Source Administrator (Windows) or the SAP Data Services Connection Manager (Unix).
- If you include SSL/TLS encryption:

- Download the SAP CommonCrypto library and set the PATH environment variable as instructed in Obtaining the SAP CommonCryptoLib file in Windows and Unix [page 53].

  > **i Note**
  >
  > If you use the SAP HANA ODBC driver version 2.9 or higher, you just have to set the PATH environment variable.

- Set SSL/TLS options in the datastore editor (Windows) or the Data Services Connection Manager (Unix).

- If you use X.509 authentication:
  - Download the SAP CommonCrypto library and set the PATH environment variable as instructed in Obtaining the SAP CommonCryptoLib file in Windows and Unix [page 53].

    > **i Note**
    >
    > If you've downloaded the SAP CommonCrypto library and set the PATH environment variable for SSL/TLS, you don't have to repeat it for X.509 authentication.

    > **i Note**
    >
    > If you use the SAP HANA ODBC driver version 2.9 or higher, you just have to set the PATH environment variable.

  - Configure the X.509 key store file. For details, see the SAP Knowledge Base article 3126555.
  - Set X.509 options in the datastore editor (Windows) or the Data Services Connection Manager (Unix).

  > **i Note**
  >
  > Support for X.509 authentication begins in SAP Data Services version 4.2 SP14 (14.2.14.16) and is applicable for SAP HANA Server on premise 2.0 SP05 revision 56 and above and SAP HANA Cloud. For SAP HANA client, X.509 is applicable for SAP HANA ODBC client 2.7 and above.

Parent topic: SAP HANA database datastores [page 59]

Related Information

SAP HANA datastore option descriptions [page 61]

3.12.5.2 SAP HANA datastore option descriptions

In addition to the common database datastore options, SAP Data Services requires that you set options specific to SAP HANA, SSL/TLS, and X.509 authentication.

For descriptions of common database datastore options, and for steps to create a database datastore, see the Datastores section of the *Designer Guide*.

The following table contains the SAP HANA-specific options in the datastore editor, including DSN and DSN-less settings.

Option	Value
Use Data Source Name (DSN)	Specifies whether to use a DSN (data source name) connection. • Select to create a datastore using a DSN. • Don't select to create a datastore using a server-name (DSN-less) connection.
The following options appear when you select to use a DSN connection:	
Data Source Name	Select the SAP HANA DSN that you created previously (see Prerequisites).
User Name *Password*	Enter the user name and password connected to the DSN. **i Note** If you use X.509 authentication, you don't have to enter a user name and password.
The following options appear when you create a DSN-less connection:	
Database server name	Specifies the name of the computer where the SAP HANA server is located. If you're connecting to SAP HANA 2.0 SPS 01 MDC or later, enter the SAP HANA database server name for the applicable tenant database. **i Note** See SAP HANA documentation to learn how to find the specific tenant database port number.
Port	Enter the port number to connect to the SAP HANA Server. The default is 30015. If you're connecting to SAP HANA 2.0 SPS 01 MDC or later, enter the port number of the specific tenant database. **i Note** See SAP HANA documentation to learn how to find the specific tenant database port number.

Advanced options

The following table contains descriptions for the advanced options in the SAP HANA datastore editor, including options for SSL/TLS and X.509.

> **i** Note
>
> Support for X.509 authentication begins in SAP Data Services version 4.2 SP14 (14.2.14.16) and is applicable for SAP HANA Server on premise 2.0 SP05 revision 56 and above, and SAP HANA Cloud. For SAP HANA client, X.509 is applicable for SAP HANA ODBC client 2.7 and above.

Option	Description
Database name	Optional. Enter the specific tenant database name. Applicable for SAP HANA version 2.0 SPS 01 MDC and later.
Additional connection parameters	Enter information for any additional parameters that the data source ODBC driver and database supports. Use the following format: `<parameter1=value1;parameter2=value2>`
Use SSL encryption	Specifies to use SSL/TLS encryption for the datastore connection to the database. • *Yes*: Creates the datastore with SSL/TLS encryption. • *No*: Creates the datastore without SSL/TLS encryption.
Encryption parameters	Opens the *Encryption Parameters* dialog box. To open the dialog, either double-click in the empty cell or select the empty cell and then select the *Ellipses* (...) icon that appears at the end. Enabled only when you select *Yes* for *Use SSL encryption* or *Yes* for *Use X.509 authentication*.
The following options are in the *Encryption Parameters* dialog box when you select *Yes* for *Use SSL encryption*:	
Validate Certificate	Specifies whether the software validates the SAP HANA server SSL certificate. If you do not select this option, none of the other SSL options are available to complete.
Crypto Provider	Specifies the crypto provider used for SSL/TLS and X.509 communication. Data Services populates *Crypto Provider* automatically with *commoncrypto*. SAP CommonCryptoLib library is the only supported cryptographic library for SAP HANA.

Option	Description
Certificate host	Specifies the host name used to verify the server identity. Choose one of the following actions: • Leave blank. Data Services uses the value in *Database server name*. • Enter a string that contains the SAP HANA server host-name. • Enter the wildcard character "*" so that Data Services doesn't validate the certificate host.
Key Store	Specifies the location and file name for your key store file. You can also use a substitution parameter. **i Note** If you choose to use X.509 authentication, the key store PSE file must contain both the SSL server certificate and X.509 client certificate. If the key store file name isn't an absolute path, Data Services assumes the PSE file is located in `$SECUDIR`.
X.509 authentication settings:	
Use X.509 authentication	Specifies to use X.509 authentication. **i Note** X.509 authentication is applicable for SAP HANA Server on premise 2.0 SP05 revision 56 and above and SAP HANA Cloud. For SAP HANA client, X.509 is applicable for SAP HANA ODBC client 2.7 and above.
X.509 key store	Specifies the x509 key store file, which includes the X.509 client certificate. If the input isn't an absolute file path, Data Services assumes that the file is in `$SECUDIR`. You can also use a substitution parameter for *X.509 key store*.
Proxy **group: For DSN-less connections only. For connecting to SAP HANA data through the SAP Cloud connector.**	
Proxy host	Specifies the proxy server host.
Proxy port	Specifies the proxy server port.
Proxy user name	Specifies the proxy server user name.
Proxy password	Specifies the proxy server password.
SAP cloud connector account	Specifies the SAP Cloud connector account. Complete when the SAP HANA data is in SAP Big Data Services.

Parent topic: SAP HANA database datastores [page 59]

Related Information

SAP HANA datastore prerequisites [page 60]

3.12.6 Configuring DSN for SAP HANA on Windows

To use a DSN connection for an SAP HANA datastore, configure a DSN connection for Windows using the ODBC Data Source Administrator.

Optionally include SSL/TLS encryption and the X.509 authentication settings when you configure the DSN.

Perform the prerequisites listed in SAP HANA datastore prerequisites [page 60].

For SSL/TLS encryption, perform the following steps:

1. Copy `sapsrv.pse` from `$SECUDIR` of the SAP HANA server.
2. Paste `sapsrv.pse` to `$SECUDIR` of the client.
3. Rename `sapsrv.pse` to `sapcli.pse`.

To configure DSN for SAP HANA on Windows, perform the following steps:

1. Open the ODBC Data Source Administrator.

   Access the ODBC Data Source Administrator either from the datastore editor in Data Services Designer or directly from your *Start* menu.
2. In the ODBC Data Source Administrator, open the *System DSN* tab and select *Add*.
3. Choose the SAP HANA ODBC driver and select *Finish*.

   The driver is listed only after you select it in the ODBC Drivers Selector utility.

   The ODBC Configuration for SAP HANA dialog box opens.
4. Enter a unique name in *Data Source Name*, and enter a description if applicable.
5. Choose the database type from the *Database type* list.
6. Enter the server host name in *Host*.
7. Enter the port number in *Port*.
8. Enter the tenant database name in *Tenant database*.
9. **Optional:** For SSL/TLS encryption and/or X.509 authentication, select *Settings*.
10. **Optional:** Complete the following options for SSL/TLS:
    a. In the *TLS/SSL* group, select the following options:
       - *Connect to the database using TLS/SSL*
       - *Validate the SSL certificate* (optional)
    b. Either leave *Certificate host* blank or enter a value:

       If you leave *Certificate host* blank, Data Services uses the value in *Database server name*. If you don't want the value from the *Database server name*, enter one of the following values:
       - A string that contains the SAP HANA server hostname.
       - The wildcard character "*", so that Data Services doesn't validate the certificate host.
    c. Specify the location and file name for your key store in *Key Store*.

d. Select *OK*.

11. **Optional:** Complete the following options for X.509 authentication:
    a. **Optional:** In the *TLS/SSL* group, select *Connect to the database using TLS/SSL*.
    b. Open *Advanced ODBC Connection Property Setup* and enter a string in *Additional connection properties*:

       To bypass certificate validation, enter the following string:

       ```
       authenticationX509:<location_of_PSE>\x509.pse
       sslValidateCertificate:FALSE
       ```

       To validate certificate, enter the following string:

       ```
       authenticationX509:<Location_of_PSE>\x509.pse
       ```

    c. Select *OK*.

Task overview: SAP HANA [page 52]

Related Information

Cryptographic libraries and global.ini settings [page 53]
X.509 authentication [page 54]
Bulk loading in SAP HANA [page 55]
Creating stored procedures in SAP HANA [page 58]
SAP HANA database datastores [page 59]
Configuring DSN for SAP HANA on Unix [page 66]
Datatype conversion for SAP HANA [page 68]
Using spatial data with SAP HANA [page 70]

3.12.7 Configuring DSN for SAP HANA on Unix

Configure a DSN connection for an SAP HANA database datastore for Unix using the SAP Data Services Connection Manager.

Optionally include SSL/TLS encryption and the X.509 authentication settings when you configure the DSN.

Perform the prerequisites listed in SAP HANA datastore prerequisites [page 60].

For SSL/TLS encryption, perform the following steps:

1. Copy `sapsrv.pse` from `$SECUDIR` of the SAP HANA server.
2. Paste `sapsrv.pse` to `$SECUDIR` of the client.
3. Rename `sapsrv.pse` to `sapcli.pse`.

Use the GTK+2 library to make a graphical user interface for the Connection Manager. Connection Manager is a command-line utility. To use it with a UI, install the GTK+2 library. 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. Export `$ODBCINI` to a file in the same computer as the SAP HANA data source. For example:

   ```
   export ODBCINI=<dir_path>/odbc.ini
   ```

2. Start SAP Data Services Connection Manager by entering the following command:

   ```
   $LINK_DIR/bin/DSConnectionManager.sh
   ```

3. Open the *Data Sources* tab and select *Add* to display the list of database types.
4. In the *Select Database Type* dialog box, select the SAP HANA database type and select *OK*.

   The configuration page opens with some of the connection information automatically completed:
   - Absolute location of the `odbc.ini` file
   - Driver for SAP HANA
   - Driver version
5. Complete the following options:

   - *DSN Name*
   - *Specify the Driver Name*
   - *Specify the Server Name*
   - *Specify the Server Instance*
   - *Specify the User Name*
   - *Type the database password*
   - *Specify the Host Name*
   - *Specify the Port*
6. **Optional:** To include SSL/TLS encryption, complete the following options:

   - *Specify the SSL Encryption Option*: Select *y*.
   - *Specify the Validate Server Certificate Option*: Select *y* or *n*.
   - *Specify the HANA SSL provider*: Accept *commoncrypto*.
   - *Specify the SSL Certificate File*: Enter the absolute path of the SSL trust store file.
   - *Specify the SSL Key File*: Enter the absolute path of the SSL key store file.

     > **i Note**
     >
     > If you enable X.509 authentication, the key store file must contain both the SSL server certificate and the X.509 client certificate.

   - *Specify the SSL Host Name in Certificate*:
     - Leave blank to use the database server name.
     - Enter a string that contains the SAP HANA server hostname.
     - Enter the wildcard character "*" so that Data Services doesn't validate the certificate host.

   For descriptions of the DSN and SSL/TLS options, see SAP HANA datastore prerequisites [page 60].
7. **Optional:** To include X.509 authentication, complete the following X.509 option: *Specify the HANA User Authentication Method*: Select *1: x.509*.
8. Press `Enter`.

The system tests the connection. The message “Successfully edited database source.” appears when you've successfully configured the DSN.

Task overview: SAP HANA [page 52]

Related Information

Cryptographic libraries and global.ini settings [page 53]
X.509 authentication [page 54]
Bulk loading in SAP HANA [page 55]
Creating stored procedures in SAP HANA [page 58]
SAP HANA database datastores [page 59]
Configuring DSN for SAP HANA on Windows [page 65]
Datatype conversion for SAP HANA [page 68]
Using spatial data with SAP HANA [page 70]

3.12.8 Datatype conversion for SAP HANA

SAP Data Services performs data type conversions when it imports metadata from SAP HANA sources or targets into the repository and when it loads data into an external SAP HANA table or file.

Data Services uses its own conversion functions instead of conversion functions that are 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, Data Services converts from internal data types to the data types of the respective DBMS.

SAP HANA datatypes [page 69]
SAP Data Services converts SAP HANA data types when you import metadata from an SAP HANA source or target into the repository.

Parent topic: SAP HANA [page 52]

Related Information

Cryptographic libraries and global.ini settings [page 53]
X.509 authentication [page 54]
Bulk loading in SAP HANA [page 55]
Creating stored procedures in SAP HANA [page 58]
SAP HANA database datastores [page 59]
Configuring DSN for SAP HANA on Windows [page 65]

Configuring DSN for SAP HANA on Unix [page 66]
Using spatial data with SAP HANA [page 70]

3.12.8.1 SAP HANA datatypes

SAP Data Services converts SAP HANA data types when you import metadata from an SAP HANA source or target into the repository.

Data Services converts data types back to SAP HANA data types when you load data into SAP HANA after processing.

Data type conversion on import

SAP HANA data type	Converts to Data Services data type
integer	int
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 type conversion on load to template table

Data Services data type	Converts to SAP HANA data type
blob	blob

Data Services data type	Converts to SAP HANA data type
date	date
datetime	timestamp
decimal	decimal
double	double
int	integer
interval	real
long	clob/nclob
real	decimal
time	time
timestamp	timestamp
varchar	varchar/nvarchar

Parent topic: Datatype conversion for SAP HANA [page 68]

3.12.9 Using spatial data with SAP HANA

SAP Data Services supports spatial data such as point, line, polygon, collection, for specific databases.

Data Services supports spatial data in the following databases:

- Microsoft SQL Server for reading
- Oracle for reading
- SAP HANA for reading and loading

When you import a table with spatial data columns, Data Services imports the spatial type columns as character based large objects (clob). The column attribute is Native Type, which has the value of the actual data type in the database. For example, Oracle is SDO_GEOMETRY, Microsoft SQL Server is geometry/geography, and SAP HANA is ST_GEOMETRY.

Limitations

- You cannot create template tables with spatial types because spatial columns are imported into Data Services as clob.
- You cannot manipulate spatial data inside a data flow because the spatial utility functions are not supported.

Loading spatial data to SAP HANA [page 71]
Load spacial data from Oracle or Microsoft SQL Server to SAP HANA.

Loading complex spatial data from Oracle to SAP HANA [page 71]
Complex spatial data is data such as circular arcs and LRS geometries.

Parent topic: SAP HANA [page 52]

Related Information

Cryptographic libraries and global.ini settings [page 53]
X.509 authentication [page 54]
Bulk loading in SAP HANA [page 55]
Creating stored procedures in SAP HANA [page 58]
SAP HANA database datastores [page 59]
Configuring DSN for SAP HANA on Windows [page 65]
Configuring DSN for SAP HANA on Unix [page 66]
Datatype conversion for SAP HANA [page 68]

3.12.9.1 Loading spatial data to SAP HANA

Load spacial data from Oracle or Microsoft SQL Server to SAP HANA.

Learn more about spatial data by reading the SAP HANA documentation.

1. Import a source table from Oracle or Microsoft SQL Server to SAP Data Services.
2. Create a target table in SAP HANA with the appropriate spatial columns.
3. Import the SAP HANA target table into Data Services.
4. Create a data flow with an Oracle or Microsoft SQL Server source as reader.
   Include any necessary transformations.
5. Add the SAP HANA target table as a loader.
   Make sure not to change the data type of spatial columns inside the transformations.
6. Build a job that includes the data flow and run it to load the data into the target table.

Task overview: Using spatial data with SAP HANA [page 70]

Related Information

Loading complex spatial data from Oracle to SAP HANA [page 71]

3.12.9.2 Loading complex spatial data from Oracle to SAP HANA

Complex spatial data is data such as circular arcs and LRS geometries.

1. Create an Oracle datastore for the Oracle table.

   For instructions, see the guide *Supplement for Oracle Applications*.

2. Import a source table from Oracle to SAP Data Services using the Oracle datastore.
3. Create a target table in SAP HANA with the appropriate spatial columns.
4. Import the SAP HANA target table into Data Services.
5. Create a data flow in Data Services, but instead of including an Oracle source, include a SQL transform as reader.
6. Retrieve the data from the Oracle database directly. First, open the SQL transform, then add the SQL Select statement. Add the SQL Select statement by calling the following functions against the spatial data column:
   - `SDO_UTIL.TO_WKTGEOMETRY`
   - `SDO_GEOM.SDO_ARC_DENSIFY`

   For example, in the SQL below, the table name is “Points”. The “geom” column contains the following geospatial data:

```
SELECT
SDO_UTIL.TO_WKTGEOMETRY(
    SDO_GEOM.SDO_ARC_DENSIFY(
        geom,
        (MDSYS.SDO_DIM_ARRAY(
        MDSYS.SDO_DIM_ELEMENT('X',-83000,275000,0.0001),
        MDSYS.SDO_DIM_ELEMENT('Y',366000,670000,0.0001)
        )),
        'arc_tolerance=0.001'
    )
)
from  "SYSTEM"."POINTS"
```

   For more information about how to use these functions, see the *Oracle Spatial Developer's Guide* on the Oracle Web page at SDO_GEOM Package (Geometry) .
7. Build a job in Data Services that includes the data flow and run it to load the data into the target table.

Task overview: Using spatial data with SAP HANA [page 70]

Related Information

Loading spatial data to SAP HANA [page 71]

3.13 SAP Vora

To access SAP Vora tables, use an 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 an SAP Vora table as a source in a data flow, and use an SAP Vora template table for the target.

SAP Data Services loads data from the Vora target template table to a CSV staging file in one of the following file types:

- Locally configured
- HDFS
- Amazon S3 HDFS

The software 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 Data Services.
- Browse metadata.

Consider the following limitations when you use an 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.

The following are 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.
- Ensure that the datastore user is registered as an SAP Vora "Vora user." For details about user types, see your SAP Vora Developer Guide.

SAP Vora datastore [page 74]
Access table data in SAP Vora using an SAP Vora datastore as a source or target in a data flow.

Configuring DSN for SAP Vora on Windows [page 74]
With SAP Vora on a Windows platform, configure a DSN type connection while you create the datastore.

Configuring DSN for SAP Vora on Unix and Linux [page 75]
With SAP Vora on Unix or Linux environments, configure a DSN type connection using the Connection Manager.

SAP Vora table source options [page 77]
Use tables imported with a SAP Vora datastore as sources in your data flows.

SAP Vora target table options [page 77]
Use an SAP Vora datastore as a target in a data flow.

SAP Vora data type conversions [page 80]
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.

3.13.1 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 table describes the options in the datastore editor specific to SAP Vora. For descriptions of the common datastore options, see the *Designer Guide*.

SAP Vora datastore options

Option	Description
Data Source Name	Specifies the data source name to use for the datastore. A DSN is required for SAP Vora datastores. 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** Ensure that you use a driver for SAP HANA version 2.0 SP2 ODBC or later version
ODBC Admin	Opens the ODBC Administrator to create a DSN.

Parent topic: SAP Vora [page 72]

Related Information

Configuring DSN for SAP Vora on Windows [page 74]
Configuring DSN for SAP Vora on Unix and Linux [page 75]
SAP Vora table source options [page 77]
SAP Vora target table options [page 77]
SAP Vora data type conversions [page 80]

3.13.2 Configuring DSN for SAP Vora on Windows

With SAP Vora on a Windows platform, configure a DSN type connection while you create the datastore.

Download and install the SAP HANA ODBC driver version 2.0 SP2 and later. Open the applicable SAP Vora datastore to open the datastore editor.

1. Click *ODBC Admin*.

   The *ODBC Data Source Administrator* dialog box opens.
2. Open the *System DSN* tab and click *Add*.

3. Select the HDBODBC driver from the list.

   The HDBODBC driver appears in the list only if you have downloaded and installed the driver as instructed in Prerequisites.
4. Click *Finish*.

   The *ODBC Configuration for SAP HANA* dialog box opens.
5. Enter a name in *Data Source Name*. Optionally enter a description in *Description*.
6. Enter the server name and port number separated with a colon in *Server:Port*.

   > **Example**
   >
   > *vora:30115*
7. If the Vora 2.x server has TLS enabled, click *Settings*.

   The *Advanced ODBC Connection Property Setup* dialog box opens.
8. Check *Connect using SSL* to enable SSL and click *OK*.
9. Click *Connect* to test the connection.
10. When the connection tests successfully, click *OK*.

Task overview: SAP Vora [page 72]

Related Information

SAP Vora datastore [page 74]
Configuring DSN for SAP Vora on Unix and Linux [page 75]
SAP Vora table source options [page 77]
SAP Vora target table options [page 77]
SAP Vora data type conversions [page 80]
Configuring DSN for SAP Vora on Unix and Linux [page 75]
SAP Vora datastore [page 74]

3.13.3 Configuring DSN for SAP Vora on Unix and Linux

With SAP Vora on Unix or Linux environments, configure a DSN type connection using the Connection Manager.

Download and install the SAP HANA ODBC driver version 2.0 SP2 and later. The file name is `libodbcHDB.so`.

Use the GTK+2 toolkit to create a graphical user interface for Connection Manager. The GTK+2 is a free multiplatform 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 GUI for Connection Manager.

1. In a Command Prompt, open the Connection Manager as follows:

Sample Code

```
$ cd $LINK_DIR/bin/
$ ./DSConnectionManager.sh
```

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.

3. Enter the remaining options as described in the following table.

   Driver options

Option	Description
ODBC ini File	Enter the absolute pathname for the `odbc.ini` file.
DSN Name	Select the name from the dropdown arrow.
User Name	Enter the user name to access the SAP Vora table.
Password	Enter the password to access the SAP Vora table.
Driver	Enter the location and name of the SAP Hana ODBC driver. Name: `libodbcHDB.so`.
Host Name	Enter the server name.
Port	Enter the port number.
SSL Encryption Option	Select *y* if Vora server has TLS enabled. Select *n* if Vora server does not have TLS enabled.

4. Optional. Click *Test Connection*. When the connection is successful, click *OK*.
5. Click *Close* to close the Connection Manager.

Task overview: SAP Vora [page 72]

Related Information

SAP Vora datastore [page 74]
Configuring DSN for SAP Vora on Windows [page 74]
SAP Vora table source options [page 77]
SAP Vora target table options [page 77]
SAP Vora data type conversions [page 80]
Configuring DSN for SAP Vora on Windows [page 74]
SAP Vora datastore [page 74]

3.13.4 SAP Vora table source options

Use tables imported with a SAP Vora datastore as sources in your data flows.

When you drag the datastore table onto the Data Services workspace as a source, the software auto completes the options in the following table. You cannot edit these options.

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.
Database type	*Vora*.

Parent topic: SAP Vora [page 72]

Related Information

SAP Vora datastore [page 74]
Configuring DSN for SAP Vora on Windows [page 74]
Configuring DSN for SAP Vora on Unix and Linux [page 75]
SAP Vora target table options [page 77]
SAP Vora data type conversions [page 80]
SAP Vora datastore [page 74]

3.13.5 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” in the *Reference Guide*.

Option tab

Option	Description
Drop and re-create table	Specifies whether the software drops the existing table and creates a different table with the same name as before loading. This option is required for SAP Vora.

Option	Description
Table type	Specifies the type of SAP Vora table to process. • *IN-MEMORY*: Uses the relational in-memory store in SAP Vora. Loads relational data into the main memory for fast access. IN-MEMORY is the default setting. Bulk loading is required for IN-MEMORY tables. If you choose IN-MEMORY, configure a valid file location supported by SAP Vora. Specify the file location in the *Vora Import File Location* option. • *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. Read about the engines in the SAP Vora *Installation and Administration Guide*.

If the existing target table is a DATASOURCE table, set the following bulk loading options.

Bulk Loader options

Option	Description
Bulk Load	Specifies whether the software uses SAP Vora bulk loading options to write data. • Selected: Enables bulk loading. • Not Selected: Disables bulk loading. **i Note** When you enable bulk loading, also set the number of loaders in *Number of Loaders* option.
Vora Import File Location	Specifies the location of the local file to use for loading SAP Vora tables. • Local • HDFS • WEBHDFS • S3 • ADL (Azure Data Lake Store) If you select *Bulk Load*, the local file system should be WEBHDFS, S3, or ADL.

Option	Description
Clean up bulk loader directory after load	Specifies whether the software deletes the data files after successfully completing the bulk load. • Selected: Deletes the data files after successfully completing the bulk load. Selected is the default. • Not selected: Does not delete the data files after successfully completing the bulk load. If the bulk load does not successfully complete, the data file and auxiliary files remain in the bulk loader directory. Ensure that you manually delete the data files. If you do not select *Bulk Load*, the files remain in the local file system for you to clean up manually.

General settings

Option	Description
Number of loaders	Specifies the number of loaders the software uses when bulk loading is enabled. • The default is *1*. • Enter more than 1 for 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. Applicable for DATASOURCE and TRANSACTIONAL table types only.

The following table describes the Vora template tables and the type of table created based on whether the *Bulk Load* option is selected or not selected.

Template table	Bulk Load option	Table type created
IN-MEMORY	Selected	DATASOURCE
DISK	Not selected (default)	STREAMING
DISK	Selected	DATASOURCE

Parent topic: SAP Vora [page 72]

Related Information

SAP Vora datastore [page 74]

Configuring DSN for SAP Vora on Windows [page 74]
Configuring DSN for SAP Vora on Unix and Linux [page 75]
SAP Vora table source options [page 77]
SAP Vora data type conversions [page 80]

3.13.6 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
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

SAP Data Services data type	SAP Vora data type
date	date
time	time
datetime	timestamp
timestamp	timestamp
blog	varchar
long	varchar

Parent topic: SAP Vora [page 72]

Related Information

SAP Vora datastore [page 74]
Configuring DSN for SAP Vora on Windows [page 74]
Configuring DSN for SAP Vora on Unix and Linux [page 75]
SAP Vora table source options [page 77]
SAP Vora target table options [page 77]
SAP Vora datastore [page 74]

4 Cloud computing services

SAP Data Services provides access to various cloud databases and storages to use for reading or loading big data.

Cloud databases [page 82]
Access various cloud databases through file location objects and file format objects.

Cloud storages [page 99]
Access various cloud storages through file location objects and gateways.

4.1 Cloud databases

Access various cloud databases through file location objects and file format objects.

SAP Data Services supports many cloud database types to use as readers and loaders in a data flow.

Amazon Redshift database [page 83]
Redshift is a cloud database designed for large data files.

Azure SQL database [page 90]
Developers and administrators who use Microsoft SQL Server can store on-premise SQL Server workloads on an Azure virtual machine in the cloud.

Google BigQuery [page 91]
The Google BigQuery datastore contains access information and passwords so that the software can open your Google BigQuery account on your behalf.

Google BigQuery ODBC [page 92]
With a Google BigQuery ODBC datastore, make ODBC calls to your Google BigQuery data sets to download, process, and upload data in SAP Data Services.

Snowflake [page 92]
Snowflake provides a data warehouse that is built for the cloud.

Parent topic: Cloud computing services [page 82]

Related Information

Cloud storages [page 99]

4.1.1 Amazon Redshift database

Redshift is a cloud database designed for large data files.

In SAP Data Services, you create a database datastore to access your data from Amazon Redshift. Additionally, load Amazon S3 data files into Redshift using the build-in function `load_from_s3_to_redshift`.

4.1.1.1 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`

The following table describes the options specific for Redshift when you create or edit a datastore:

Main window options

Option	Description
Enable Automatic Data Transfer	Select to enable transfer tables in this datastore. The Data_Transfer transform uses transfer tables to push down subsequent database operations. This option is enabled by default.

Option	Description
Advanced > *Connection* > *Additional connection parameters*	When setting up an Amazon Redshift ODBC driver connection, used to insert the Boolean data type as an integer. When creating a server-based connection to a Redshift cluster database, utilized when you want to use a server certificate. For details about both, see the information below.

You can 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.

Be sure to 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*.

> **i** Note
>
> The Redshift ODBC driver sets the bool data type to `string` by default, but you cannot insert a string to the Bool data type column on the latest ODBC driver. Therefore, you must insert an integer into the Bool column.

Creating a Server-Based Connection

You can create a server-based connection to connect to the Redshift cluster database. To do so, you must do the following:

- Before using this connection, select the Redshift ODBC driver by using `ODBCDriverSelector.exe` under `%link_dir%\bin`. When you create this new datastore, make sure that *Use data source name (DNS)* is not selected.
- Because the Redshift ODBC driver sets the Boolean data type to `string` by default, you must import the Boolean data type as an integer by adding the Advanced Connection configuration option *Additional connection parameters* of **`BoolsAsChar=0`** and then re-import the table. The Bool data type column will then be `int`
- If using a DSN connection, uncheck the data type option *Show Boolean Colum As String* during configuration.
- The default SSL option of the server-based datastore connection *SSLMode* sets as required. If you use a server certificate (`verify-ca` and `verify-full`), you must change the SSL option by adding it manually in the datastore's Advanced Connection configuration option *Additional connection parameters*. For example, `SSLMode=<Redshift SSL option>`.
  If the `root.crt` file is not in your ODBC driver library directory, for example, on Windows C:\Program Files \Amazon Redshift ODBC Driver\lib, you must download the server certificate from Amazon and save the certificate as `root.crt` in the ODBC library directory, or save the server certificate in a special directory on your system. If you save the server certificate in a special directory, you need add the certificate path in the datastore's *Additional connection parameters* connection option, such as `SSLCertPath=<<full path of SSL certificate including certificate file name>>;`.

For details about Amazon Redshift support, see the *Supplement for Big Data*.

4.1.1.2 Configuring a Redshift connection on Linux using ODBC DNS

Use the DSConnection Manager to configure Amazon Redshift as a source for Data Services that uses an ODBC data source name connection.

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*.
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.

   > **i 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.
```

4.1.1.3 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 about push down functions, see SAP Note 2212730, "SAP Data Services push-down operators, functions, and transforms". Also read about 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
Table name	Name of the table that you added as a source to the data flow.
Table owner	Owner that you entered when you created the Redshift table.
Datastore name	Name of the Redshift datastore.
Database type	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.

For more information about pushdown operations and viewing optimized SQL, see the *Performance Optimization Guide*.

Related Information

Amazon Redshift data types [page 89]

Amazon Redshift target table options [page 87]
Amazon Redshift datastores [page 83]

4.1.1.4 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.

> **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.

> **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
Bulk load	Select to use bulk loading options to write the data.
Mode	Select the mode for loading data in the target table: • *Append*: Adds new records to the table. > **Note** > Append mode does not apply to template tables. • *Truncate*: Deletes all existing records in the table, and then adds new records.
S3 file location	Enter or select the path to the Amazon S3 configuration file. You can enter a variable for this option.

Option	Description
Maximum rejects	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.
Column delimiter	Enter a single-character column delimiter.
Text delimiter	Enter a single-character text delimiter. If you insert a *Text delimiter*, other than a single quote (‘), as well as a comma (,) for the *Column delimiter*, Data Services will treat the data file as a `.csv` file.
Generate files only	Enable to generate data files that you can use for bulk loading. 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. If you do not specify a bulk loader directory, the software writes the files to `<%DS_COMMON_DIR%>\log\bulkloader\<tablename><PID>`. Then you manually copy the files to the Amazon S3 remote system. The file name is `<tablename><PID>_<timestamp>_<loader_number>_<number of files generated by each loader>.dat`, where `<tablename>` is the name of the target table.
Clean up bulk loader directory after load	Enable to delete all bulk load-oriented files from the bulk load directory and the Amazon S3 remote system after the load is complete.
Parameters	Allows you to enter some Amazon Redshift copy command data conversion parameters, such as escape, emptyasnull, blanksasnull, ignoredblanklines, 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
Number of loaders	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

Amazon Redshift source [page 86]
Amazon Redshift datastores [page 83]

4.1.1.5 Amazon Redshift data types

SAP Data Services converts Redshift data types to the internal data types when it imports metadata from a Redshift source or target into the repository.

The following table lists the internal data type that Data Services uses in place of the Redshift data type.

Convert Redshift to Data Services internal data type

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 **i Note** The char data type doesn't support multibyte characters. The maximum range is 4096 bytes.
nchar	char
varchar nvarchar	varchar **i Note** The varchar and nvarchar data types support UTF8 multibyte characters. The size is the number of bytes and the max range is 65535. **⚠ Caution** If you try to load multibyte characters into a char or nchar data type column, Redshift issues an error. Redshift internally converts nchar and nvarchar data types to char and varchar. The char data type in Redshift doesn't support multibyte characters. Use overflow to catch the unsupported data or, to avoid this problem, create a varchar column instead of using the char data type.
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 use a Redshift template table as the target.

Data Services internal data type to Redshift data type

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

4.1.2 Azure SQL database

Developers and administrators who use Microsoft SQL Server can store on-premise SQL Server workloads on an Azure virtual machine in the cloud.

The Azure virtual machine supports both Unix and Windows platforms.

Moving files to and from Azure containers [page 91]

To move blobs to or from a Microsoft Azure container, use built-in functions in a script object and a file location object.

4.1.2.1 Moving files to and from Azure containers

To move blobs to or from a Microsoft Azure container, use built-in functions in a script object and a file location object.

> **i Note**
>
> Files are called blobs when they are in an Azure container.

Before you perform the following steps, create a file location object for the Azure container. Also create an unstructured binary text file format that describes the blob in the Azure container. Use the file format in a data flow to perform extra operations on the blob.

Use an existing Azure container or create a new one. Because SAP Data Services doesn't internally manipulate the blobs in an Azure container, the blobs can be of any type. Currently, Data Services supports the block blob in the container storage type.

To use built-in functions to upload files to a container storage blob in Microsoft Azure, perform the following high-level steps:

1. Create a storage account in Azure and take note of the primary shared key.

   For instructions, see your Microsoft documentation.
2. Create a file location object with the Azure Cloud Storage protocol.
3. Create a job in Data Services Designer and include the applicable file format as source or target.
4. Add scripts to the job that contain one or both of the following built-in functions:

   To move files between remote and local directories, use the following scripts:

   - `copy_to_remote_system`
   - `copy_from_remote_system`

   > **Example**
   >
   > To access a subfolder in your Azure container, specify the subfolder in the following script:
   >
   > ```
   > copy_to_remote_system('New_FileLocation', '*', '<container_name>/
   > <remote_directory>/<sub_folder>')
   > ```
   >
   > The script copies all of the files from the local directory specified in the file location object to the container specified in the same object. When you include the remote directory and subfolder in the script, the function copies all of the files from the local directory to the subfolder specified in the script.
5. Save and run the job.

4.1.3 Google BigQuery

The Google BigQuery datastore contains access information and passwords so that the software can open your Google BigQuery account on your behalf.

After accessing your account, SAP Data Services can load data to or extract data from your Google BigQuery projects:

- Extract data from a Google BigQuery table to use as a source for Data Services processes.
- Load generated data from Data Services to Google BigQuery for analysis.
- Automatically create and populate a table in your Google BigQuery dataset by using a Google BigQuery template table.

You can also reference the Google BigQuery datastore in the built-in function `load_from_gcs_to_gbq`, to load data from your GCS to a Google BigQuery table. For details about the function, see the *Reference Guide*.

> **i Note**
>
> Beginning with Data Services 4.2.13, we recommend that you create a Google BigQuery ODBC datastore instead of the Google BigQuery datastore. The Google BigQuery ODBC datastore uses the Magnitude Simba ODBC driver for BigQuery, which supports standard SQL and more data types than the Google BigQuery datastore.

For complete information about how Data Services supports Google BigQuery, see the *Supplement for Google BigQuery*.

4.1.4 Google BigQuery ODBC

With a Google BigQuery ODBC datastore, make ODBC calls to your Google BigQuery data sets to download, process, and upload data in SAP Data Services.

To access the data in your Google BigQuery account, the datastore uses the Simba ODBC driver for Google BigQuery, which supports the OAuth 2.0 protocol for authentication and authorization. Configure the driver to provide your credentials and authenticate the connection to the data using either a Google user account or a Google service account.

> **i Note**
>
> Beginning with Data Services 4.2.13, we recommend that you create a Google BigQuery ODBC datastore instead of the Google BigQuery datastore. The Google BigQuery ODBC datastore uses the Magnitude Simba ODBC driver for BigQuery, which supports standard SQL and more data types than the Google BigQuery datastore.

> **i Note**
>
> Data Services supports Google BigQuery ODBC datastore on Windows and Linux platforms only.

For information about how Data Services supports Google BigQuery ODBC, see the *Supplement for Google BigQuery*.

4.1.5 Snowflake

Snowflake provides a data warehouse that is built for the cloud.

After connecting to Snowflake, you can do the following:

- Import tables
- Read or load Snowflake tables in a data flow
- Create and load data into template tables
- Browse and import the tables located under different schemas (for example, Netezza)
- Preview data
- Push down base SQL functions and Snowflake-specific SQL functions (see SAP Note 2212730)
- Bulkload data (possible through AWS S3 File Location or Azure Cloud Storage File Location)

For instructions to create a target template table, see the Template tables section of the *Designer Guide*.

Using an ODBC driver to connect to Snowflake [page 93]
Use the DSConnection Manager to configure Snowflake as a source for Data Services.

Snowflake source [page 94]
Option descriptions for using a Snowflake database table as a source in a data flow.

Snowflake data types [page 96]
SAP Data Services converts Snowflake data types to the internal data types when it imports metadata from a Snowflake source or target into the repository.

Snowflake target table options [page 97]
When you use a Snowflake table as a target in SAP Data Services, complete the target options.

4.1.5.1 Using an ODBC driver to connect to Snowflake

Use the DSConnection Manager to configure Snowflake as a source for Data Services.

Before you perform the following steps, read the following information in the *Administrator Guide*:

- Configure drivers with data soure name (DSN) connections
- Using the ODSBC Drivers Selector for Windows

For steps to define a database datastore, see the *Designer Guide*.

To use an ODBC driver to connect to Snowflake, perform the following steps:

1. Download and install the Snowflake ODBC driver from the Snowflake website.

   Data Services supports Snowflake ODBC Driver version 2.16.0 and higher.

   For more information about the Snowflake ODBC driver, see the *Snowflake User Guide* on the Snowflake website.
2. Configure a Data Source Name (DSN).

   For Windows:

   a. Open *ODBC Data Source Administrator* from your Windows *Start* menu or click the *ODBC Administrator* button in the *Datastore Editor* when you create the Snowflake datastore in Data Services.
   b. In the *ODBC Administrator*, open the *System DSN* tab and select the ODBC driver for Snowflake that you just installed.
   c. Click *Configure*.
   d. Enter the required information into the *Snowflake Configuration Dialog* window and click *Save*. For information about the connection parameters, see the *Snowflake User Guide*.

For Linux:

a. Run the Connection Manager utility: `DSConnectionManager.sh`.

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

b. Type the number that corresponds to the database type.

   For example, type *18* for Snowflake.

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

   For information about the connection parameters, see the *Snowflake User Guide*.

3. After you have configured the Snowflake 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`:

   ```
   Driver=/usr/lib64/snowflake/odbc/lib/libSnowflake.so
   UIS=sapdhuser
   PWD=
   SERVER=<server URL>
   PORT=<port number>
   DATABASE=DS-DB
   SCHEMA=
   WAREHOUSE=
   ROLE=
   ```

   The `ds_odbc.ini` file is located in `<LINK_DIR>/bin`:

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

4. Create a Snowflake database datastore using the DSN you just created. For more information, see "Defining a database datastore" in the Designer Guide.

4.1.5.2 Snowflake source

Option descriptions for using a Snowflake database table as a source in a data flow.

When you use a Snowflake table as a source, the software supports the following features:

- All Snowflake data types
- SQL functions and snowflake-specific SQL function
- Push-down ODBC generic functions

For more about push down functions, see SAP Note 2212730, "SAP Data Services push-down operators, functions, and transforms". Also read about maximizing push-down operations in the *Performance Optimization Guide*.

The following table lists source options when you use a Snowflake table as a source:

Option	Description
Table name	Name of the table that you added as a source to the data flow.
Table owner	Owner that you entered when you created the Snowflake table.
Datastore name	Name of the Snowflake datastore.
Database type	Database type that you chose when you created the datastore. You cannot change this option.
Table Schema	Name of the table schema.

Performance settings

Option	Description
Join Rank	Specifies the rank of the data file relative to other tables and files joined in a data flow. Enter a positive integer. The default value is *0*. The software joins sources with higher join ranks before joining sources with lower join ranks. If the data flow includes a Query transform, the join rank specified in the *Query* transform overrides the *Join Rank* specified in the *File Format Editor*. For new jobs, specify the cache only in the Query transform editor. For more information about setting "Join rank", see "Source-based performance options" in the *Performance Optimization Guide*.
Cache	Specifies whether the software reads the data from the source and load it into memory or pageable cache. • *Yes*: Always caches the source unless the source is the outer-most source in a join. *Yes* is the default setting. • *No*: Never caches the source. If the data flow includes a Query transform, the cache setting specified in the *Query* transform overrides the *Cache* setting specified in the *Format File Editor* tab. For new jobs, specify the cache only in the Query transform editor. For more information about caching, see "Using Caches" in the *Performance Optimization Guide*.

Option	Description
Array fetch size	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.

4.1.5.3 Snowflake data types

SAP Data Services converts Snowflake data types to the internal data types when it imports metadata from a Snowflake source or target into the repository.

The following table lists the internal data type that Data Services uses in place of the Snowflake data type.

Snowflake data type	Converts to Data Services data type	Notes
byteint/tinyint	decimal(38,0)	
smallint	decimal(38,0)	
int/integer	decimal(38,0)	
bigint	decimal(38,0)	
number/numeric/decimal	decimal	Default of precision is 38.
float	double	
double	double	
real	double	
varchar	varchar	
char	varchar	Default is 1 byte.
string/text	varchar	Default is 16 mbyte.
boolean	int	
binary	blob	
varbinary	blob	
datetime/timestamp	datetime	
date	date	
time	time	
semi-structure	not supported	VARIANT, OBJECT, ARRAY

If Data Services encounters a column that has an unsupported data type, it does not import the column. However, you can configure Data Services to import unsupported data types by checking the *Import unsupported data types as VARCHAR of size* option in the datastore editor dialog box.

4.1.5.4 Snowflake target table options

When you use a Snowflake table as a target in SAP Data Services, complete the target options.

The Snowflake target supports the following features:

- transactional loads
- load triggers, pre-load commands, and post-load commands
- bulk loading

After you add the Snowflake table as a target in a data flow, open the *Bulk loader* tab and complete the options as described in the following table.

Bulk loader tab

Option	Description
Bulk load	Select to use bulk loading options to write the data.
Mode	Select the mode for loading data in the target table: • *Append*: Adds new records to the table. **i Note** Append mode does not apply to template tables. • *Truncate*: Deletes all existing records in the table, and then adds new records.
Remote Storage	Select the remote storage method: • *Amazon S3*: Utilizes the S3 file location to copy the local data file into the staging storage temporarily and then load it to the target Snowflake table. Both the staging local file and the duplicated one in the staging storage can be cleaned according to the setting to the bulk loader. • *Microsoft Azure*: Utilizes the Azure file location to copy the local data file into the staging storage temporarily and then load it to the target Snowflake table. Both the staging local file and the duplicated one in the staging storage can be cleaned according to the setting to the bulk loader.
File Location	Enter or select the corresponding Amazon S3 or Microsoft Azure file location. You can enter a variable for this option.

Option	Description
Generate files only	Enable to generate data files that you can use for bulk loading. 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. If you do not specify a bulk loader directory, the software writes the files to `%DS_COMMON_DIR%/log/BulkLoader/<filename>` . Then you manually copy the files to the Amazon S3 or Microsoft Azure remote system. The file name is `SFBL_<DatastoreName>_<DBname>_<SchemaName>_<TableName>_***.dat`, where `<TableName>` is the name of the target table.
Clean up bulk loader directory after load	Enable to delete all bulk load-oriented files from the bulk load directory after the load is complete.

Complete the General settings as described in the following table:

Option	Description
Column comparison	Specifies how the software maps the input columns to persistent cache table columns. • *Compare by position*: The software 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. Compare by name is the default setting.
Number of loaders	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).

Also complete error handling and transaction control options as described in the following topics in the *Reference Guide*:

- Error Handling options
- Transaction control

4.2 Cloud storages

Access various cloud storages through file location objects and gateways.

File location objects specify specific file transfer protocols so that SAP Data Services safely transfers data from server to server.

For information about the SAP Big Data Services to access Hadoop in the cloud, see *Supplement for Big Data Services*.

Amazon S3 [page 99]
: Amazon Simple Storage Service (S3) is a product of Amazon Web Services that provides scalable storage in the cloud.

Azure Blob Storage [page 104]
: In SAP Data Services Designer, we refer to Azure Blob Storage as **Azure Cloud Storage**.

Azure Data Lake Storage [page 109]
: Use an Azure Data Lake Storage file location object to download data from and upload data to your Azure Data Lake Storage.

Google Cloud Storage file location [page 112]
: A Google Cloud Storage (GCS) file location contains file transfer protocol information for moving large data files, 10 MB and larger, between GCS and SAP Data Services.

Parent topic: Cloud computing services [page 82]

Related Information

Cloud databases [page 82]

Upload data to HDFS in the cloud [page 14]

4.2.1 Amazon S3

Amazon Simple Storage Service (S3) is a product of Amazon Web Services that provides scalable storage in the cloud.

Store large volumes of data in an Amazon S3 cloud storage account. Then use SAP Data Services to securely download your data to a local directory. Configure a file location object to specify both your local directory and your Amazon S3 directory.

Data Services provides built-in functions for processing data that you can use with data from S3 and data that you load to S3. There is one built-in function specifically for moving data from S3 to Amazon Redshift named `load_from_s3_to_redshift`.

Also use the `copy_to_remote_system` and `copy_from_remote_system` functions. Data Services concatenates the remote directory that you specify in the copy function with the information in the file location object to form a full directory structure that includes subfolders.

For descriptions of all built-in functions, see the *Reference Guide*.

Amazon S3 file location protocol options [page 100]
When you configure a file location object for Amazon S3, complete all applicable options, especially the options specific to Amazon S3.

4.2.1.1 Amazon S3 file location protocol options

When you configure a file location object for Amazon S3, complete all applicable options, especially the options specific to Amazon S3.

Use a file location object to access data or upload data stored in your Amazon S3 account. To view options common to all file location objects, see the *Reference Guide*. The following table describes the file location options that are specific to the Amazon S3 protocol.

> **!** Restriction
>
> You must have "s3:ListBucket" rights in order to view a list of buckets or a special bucket.

Option	Description
Access Key	Specifies the Amazon S3 identification input value.
Secret Key	Specifies the Amazon S3 authorization input value.
Region	Specifies the name of the region you are transferring data to and from; for example, "South America (Sao Palo)".
Server-Side Encryption	Specifies the type of encryption method to use. 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: • *None* • *Amazon S3-Managed Keys* • *AWS KMS-Managed Keys* • *Customer-Provided Keys* Data Services displays either one or none of the three remaining encryption options based on your selection here.
Encryption Algorithm	Specifies the encryption algorithm to use to encode the data. For example `AES256,aws:kms`.

Option	Description
AWS KMS Key ID	Specifies whether to create and manage encryption keys via the Encryption Keys section in AWS IAW console. Leave this option blank to use a default key that is unique to you, the service you're using, and the region in which you're working.
AWS KMS Encryption Context	Specifies the encryption context of the data. The value is a base64-encoded UTF-8 string holding JSON with the encryption context key-value pairs. **Example** If the encryption context is {"fullName": "John Connor" }, you need base64-encoded: **`echo {"fullName": "John Connor" } | openssl enc -base64`** **`eyJmdWxsTmFtZSI6ICJKb2huIENvbm5vciIgfSANCg==`** Enter **`eyJmdWxsTmFtZSI6ICJKb2huIENvbm5vciIgfSANCg==`** in the encryption context option.
Customer Key	Specifies the key. Enter a value less-than or equal-to 256 bits.

Option	Description
Communication Protocol/Endpoint URL	Specifies the communication protocol you use with S3. • *http* • *https* • Enter the endpoint URL If you choose to enter the endpoint URL, consider the following information: • If the endpoint URL is for http, and it contains a region, ensure that you use a dash before the region. **Example** For example, enter `http://s3-<region>.amazonaws.com`. For the U.S. East (N.Virginia) region, the endpoint is `http://s3.amazonaws.com`. Notice the period instead of a dash. • If the endpoint URL is for https, enter the endpoint URL using either a dash or a period. **Example** Enter either `https://s3-<region>.amazonaws.com` or `https://s3.<region>.amazonaws.com`.
Compression Type	Specifies the compression type to use. The software compresses the files before uploading to S3 and decompresses the files after download from S3. **Note** When you upload a file to the Amazon S3 cloud server using the `copy_to_remote_system()` function and gzip compression, Data Services adds a `.gz` file extension to the file name. For example, `sample.txt.gz`. Read about the `copy_to_remote_system()` built-in function in the *Reference Guide*. When you download the file, Data Services decompresses the file and removes the `.gz` extension from the file name. For example, `sample.txt`.

Option	Description
Connection Retry Count	Specifies the number of times the software should try to upload or download data before stopping the upload or download.
Batch size for uploading data, MB	Specifies the size of the data transfer to use for uploading data to S3. 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.
Batch size for downloading data, MB	Specifies the size of the data transfer Data Services uses to download data from S3.
Number of threads	Specifies the number of upload and download threads for transferring data to S3.
Storage Class	Specifies the S3 cloud storage class to use to restore files. • *STANDARD*: Default storage class. • *REDUCED_REDUNDANCY*: For noncritical, reproducible data. • *STANDARD_IA*: Stores object data redundantly across multiple geographically separated availability zones. • *ONEZONE_IA*: Stores object data in only one availability zone. **Note** The GLACIER storage class is not supported. Data Services can't specify GLACIER storage class during object creation. For more information about the storage classes, see the Amazon AWS documentation.
Remote directory	Optional. Specifies the name of the directory for Amazon S3 to transfer files to and from.
Bucket	Specifies the name of the Amazon S3 bucket that contains the data.
Local directory	Optional. Specifies the name of the local directory to use to create the files. If you leave this field empty, Data Services uses the default Data Services workspace.
Proxy host, port, user name, password	Specifies proxy information when you use a proxy server.

For information about `load_from_s3_to_reshift` and other built-in functions, see the Reference Guide.

For more information about file location common options, see the *Reference Guide*.

Related Information

Amazon Redshift datastores [page 83]

4.2.2 Azure Blob Storage

In SAP Data Services Designer, we refer to Azure Blob Storage as **Azure Cloud Storage**.

To access your Azure Blob Storage from Data Services, create an Azure Cloud Storage file location object.

Azure Cloud Storage file location protocol [page 104]
: Use a file location object to download data from or upload data to your Azure Blob Storage account.

Number of threads for Azure blobs [page 108]
: The number of threads is the number of parallel uploaders or downloaders to be run simultaneously when you upload or download blobs.

Related Information

Azure Data Lake Storage [page 109]

4.2.2.1 Azure Cloud Storage file location protocol

Use a file location object to download data from or upload data to your Azure Blob Storage account.

The following table lists the file location object descriptions for the Azure Cloud Storage protocol. To view options common to all file location objects, see the *Reference Guide*.

> **i** Note
>
> To work with Azure Blob data in your Data Lake Storage account, see Azure Data Lake Storage [page 109].

Option	Description
Name	Enter a unique name for the file location object.
Protocol	Choose *Azure Cloud Storage* for the type of file transfer protocol.
Account Name	Enter the name for the Azure Data Lake Storage account.

Option	Description
Storage Type	Choose the storage type to access. Data Services supports only one type for Azure Blob Storage: *Container*.
Authorization Type	Choose the type of storage access signature (SAS) authorization for Azure storage. • *Shared Key*: Account level SAS for Azure Storage Services. Accesses resources in one or more storage services. • *File (Blob) Shared Access Signature*: Service level SAS for Azure blob storage services. Accesses a specific file (blob). **i Note** *File (Blob) Shared Access Signature* isn't applicable for Snowflake bulk loading. Snowflake bulk loading requires file names that are generated randomly at runtime. Therefore, select *Shared Key* or *Container Shared Access Signature* for Snowflake with bulk loading • *Container Shared Access Signature*: Service level SAS for Azure container storage services. Accesses files (blobs) in a container.
Shared Access Signature URL	Enter the access URL that enables access to a specific file (blob) or blobs in a container. Azure recommends that you use HTTPS instead of HTTP. **i Note** Applicable when you select *File (Blob) Shared Access Signature* or *Container Shared Access Signature* for *Authorization Type*. To access blobs in a container, include the following elements in the URL: `https://<storage_account_name>/<container_name>/<signature value>` To access a specific file (blob), include the following elements in the URL: `https://<storage_account_name>/<container_name>/<file_name>/<signature value>`

Option	Description
Account Shared Key	Specify the Account Shared Key. Obtain a copy from the Azure portal in the storage account information. **i Note** For security, the software doesn't export the account shared key when you export a data flow or file location object that specifies *Azure Cloud Storage* as the protocol.
Connection Retry Count	Specify the number of times the computer tries to create a connection with the remote server after a connection fails. The default value is **10**. The value can't be zero. After the specified number of retries, Data Services issues an error message and stops the job.
Batch size for uploading data, MB	Specify the maximum size of a data block per request when transferring data files. The limit is 4 MB. **⚠ Caution** Accept the default setting unless you're an experienced user with an understanding of your network capacities in relation to bandwidth, network traffic, and network speed.
Batch size for downloading data, MB	Specify the maximum size of a data range to be downloaded per request when transferring data files. The limit is 4 MB. **⚠ Caution** Accept the default setting unless you're an experienced user with an understanding of your network capacities in relation to bandwidth, network traffic, and network speed.
Number of threads	Specify the number of upload and download threads for transferring data to Azure Cloud Storage. The default value is 1. 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 108].

Option	Description
Remote Path Prefix	Optional. Specify the file path for the remote server, excluding the server name. You must have permission to this directory. If you leave this option blank, the software assumes that the remote path prefix is the user home directory used for FTP. 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. 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 Container type storage is a flat file storage system and it doesn't 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. **Example** 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".
Local Directory	Specify the path of your local server directory for the file upload or download. Requirements for local server: • must exist • located where the Job Server resides • you have appropriate permissions for this directory For Azure Blob Storage location only: 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. Not applicable for FTP. 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.

Option	Description
Container	Specify the Azure container name for uploading or downloading blobs to your local directory. If you specified the connection information, including account name, shared key, or proxy information, select the *Container* field. The software requests a list of existing containers from the server 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.
Proxy Host, Port, User Name, Password	Optional. Specify the proxy information when you use a proxy server.

4.2.2.2 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.

4.2.3 Azure Data Lake Storage

Use an Azure Data Lake Storage file location object to download data from and upload data to your Azure Data Lake Storage.

For information about using variables and parameters to increase the flexibility of jobs, work flows, and data flows, see the *Designer Guide*.

Azure Data Lake Storage Gen1

Use Azure Data Lake Storage Gen1 for big data analytic processing. To create a file location object for Gen1, make sure that you select *Gen1* for the *Version* option.

Azure Data Lake Storage Gen2

Azure Data Lake Storage Gen2 has all of the capabilities of Gen1 plus it's built on Azure Blob storage. To create a file location object for Gen2, make sure that you select *Gen2* for the *Version* option.

Azure Data Lake Storage file location object options

The following table describes the file location options for the Azure Data Lake Storage protocol. The table combines options for Azure Data Lake Storage Gen1 and Gen2. The table contains a **Version** column that indicates the applicable version for the option.

> **i** Note
>
> For descriptions of the common file location options, see the *Reference Guide*.

Option	Description	Version
Name	Enter a name for the file location.	Both
Protocol	Select *Azure Data Lake Store*. Optionally, choose a substitution parameter.	Both
Version	Select *Gen1* or *Gen2*.	Both
Authorization Type	Select *Shared Key* or *Service Principal*.	Gen2
Account Shared Key	When *Authorization Type* is set to *Shared Key*, enter the account shared key that you obtain from your administrator.	Gen2
Communication Protocol/Endpoint URL	HTTPS or the endpoint URL.	Gen2
Data Lake Store name	Name of the Azure Data Lake Storage to access. Optionally use a substitution parameter.	Both
Service Principal ID	Obtain from your Azure Data Lake Storage administrator.	Gen1
Tenant ID	Obtain from your Azure Data Lake Storage administrator.	Gen1
Password	Obtain from your Azure Data Lake Storage administrator.	Gen1

Option	Description	Version
Batch size for uploading data (MB)	Maximum size of a data block to upload per request when transferring data files. • **Gen1:** Default value is 4 MB. Maximum value is 4 MB. • **Gen2:** Default value is 10 MB. Maximum value is 100 MB. ⚠ Caution Keep the default setting unless you're an experienced user with an understanding of your network capacities in relation to bandwidth, network traffic, and network speed.	Both
Batch size for downloading data (MB)	Maximum size of a data range to download per request when transferring data files. • **Gen1:** Default value is 4 MB. Maximum value is 4 MB. • **Gen2:** Default value is 10 MB. Maximum value is 100 MB. ⚠ Caution Keep the default setting unless you're an experienced user with an understanding of your network capacities in relation to bandwidth, network traffic, and network speed.	Both
Number of threads	Number of parallel uploaders or downloaders to run simultaneously. The default value is 1. → Tip Setting the *Number of threads* to more than 1 can reduce the time to upload Azure blob storage to or download azure blob storage from Azure Data Lake Storage.	Both

Option	Description	Version

Remote path prefix	Directory path for your files in the Azure Data Lake Storage. Obtain the directory path from Azure Data Lake Storage Properties.
	
	Example
	If the following is the directory path for your Azure Data Lake Storage Properties:
	`adl://<yourdatastoreName>.azuredatalakestore.net/<FolderName>/<subFolderName>`
	Then the *Remote path prefix* value is:
	`<FolderName>/<subFolderName>`
	
	Permission to access the directory is required.
	
	Optionally use a substitution parameter.
		Both

Local directory	Path to the local directory for your local Data Lake Storage data. Permission to access this directory required. Optionally use a substitution parameter.	Both
Container	Container name. Must be lower case and have a length of more than 3 characters.	Gen2

4.2.4 Google Cloud Storage file location

A Google Cloud Storage (GCS) file location contains file transfer protocol information for moving large data files, 10 MB and larger, between GCS and SAP Data Services.

To work with your data from the Google Cloud Platform, create a file location object that contains your account connection information and Google file transfer protocol. Use the file location in the following ways:

- As a source, select the GCS file location name in the *Cloud Storage File Location object* option in a Google BigQuery datastore configuration.
- As a target, enter the location and file name for the GCS file location object in the *Location* or *File location* option in a target editor.

You can also use the information from the GCS file location in the built-in function `load_from_gcs_to_gbq`, to load data from your GCS to a Google BigQuery table. The function includes the Google BigQuery datastore, which names the GCS file location. For details about the function, see the *Reference Guide*.

Uploading to GCS can use a large amount of local disk space. When you upload data to GCS in flat files, XML files, or JSON files, consider setting the number of rows in the target editor option *Batch size (rows)*. The option can reduce the amount of local disk space that Data Services uses to upload your data.

> **Note**
>
> If you set *Location* or *File location* to a location and file name for a non-GCS file location object, Data Services ignores the setting in *Batch size (rows)*.

To learn about file location objects, and to understand how SAP Data Services uses the file transfer protocol in the file location configuration, see the *Reference Guide*.

GCS file location option descriptions [page 113]
Complete the options in a Google Cloud Storage (GCS) file location to be able to extract data from and load data to your GCS account.

Decrease local disk usage when uploading [page 119]
Decrease the local disk space used when SAP Data Services uploads a large file of generated data to your Google Cloud Storage (GCS) account.

4.2.4.1 GCS file location option descriptions

Complete the options in a Google Cloud Storage (GCS) file location to be able to extract data from and load data to your GCS account.

The GCS file location contains connection and access information for your GCS account. The following table lists the file location object descriptions for the Google Cloud Storage file location. For descriptions of common options, and for more information about file location objects, see the *Reference Guide*.

GCS file location option descriptions

Option	Description
Name	Specifies a unique name for the file location object.
Protocol	Specifies the type of file transfer protocol. Select *Google Cloud Storage*.
Project	Specifies the Google project name to access.
Upload URL	Specifies the URL for uploading data to GCS. Accept the default, which is `https://www.googleapis.com/upload/storage/v1`.
Download URL	Specifies the URL for extracting data from GCS. Accept the default, which is `https://www.googleapis.com/storage/v1`.
Authentication Server URL	Specifies the Google server URL plus the name of the Web access service provider, which is OAuth 2.0. Accept the default, which is `https://accounts.google.com/o/oauth2/token`.

Option	Description
Authentication Access Scope	Specifies the specific type of data access permission. • *Read-only*: Access for reading data, including listing buckets. • *Read-write*: Access for reading and changing data. Not applicable for metadata like access control lists (ACLs). • *Full-control*: Access with full control over data, including the ability to modify ACLs. • *Cloud-platform.read-only*: Access for viewing your data across Google Cloud Platform services. For Google Cloud Storage, this option is the same as `devstorage.read-only`. • *Cloud-platform*: Access for viewing and managing data across all Google Cloud Platform services. For Google Cloud Storage, this option is the same as `devstorage.full-control`. *Cloud-platform*: is the default setting.
OAuth Mechanism	Specifies the authentication mechanism. • *Service Authentication* (default): Data Services holds the credentials of a service account to complete authentication. • *User Authentication*: Data Services obtains credentials from the end user. You are required to sign into Google to complete the authentication if you don't have a refresh token. You can also use an existing refresh token. When *User Authentication* is specified, you must also populate *Client ID*, *Client Secret*, and *Refresh Token*.
Service Account Email Address	Specifies 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 Note** Obtain the e-mail address when you sign in to your Google project.

Option	Description
Service Account Private Key	Specifies the P12 or JSON file you generated from your Google project and stored locally. **Note** Log in to your Google project and select to generate the private key file in either P12 or JSON format, and save the file to your local computer. Click *Browse* and open the location where you saved the file. Select the .p12 or .JSON file and click *Open*.
Service Account Signature Algorithm	Specifies the algorithm type that Data Services uses to sign JSON Web tokens. Accept the default, which is **SHA256withRSA**. Data Services uses this value, along with your service account private key, to obtain an access token from the Authentication Server.
Substitute Access Email Address	Optional, for Google BigQuery. Enter the substitute e-mail address from your Google BigQuery datastore.
Client ID	Specifies the OAuth client ID for Data Services. To get the client ID, go to the Google API Console (https://console.developers.google.com/). For more detailed instructions, see the Google documentation.
Client Secret	Specifies the OAuth client secret for Data Services. To get the client secret, go to the Google API Console (https://console.developers.google.com/). For more detailed instructions, see the Google documentation.

Option	Description
Refresh Token	Specifies the refresh token that is required for the Google Cloud Storage file location connection when *OAuth mechanism* is set to *User Authentication*. To enter an existing refresh token, do the following: 1. Click the ellipsis (...) next to the *Refresh Token* field. 2. In the *Get Refresh Token* window, select *Refresh Token*. 3. Click *OK*. The refresh token appears (masked for privacy and security reasons) in the *Refresh Token* field . To generate a new refresh token, enter an authorization code. 1. Click the ellipsis (...) next to the *Refresh Token* field. 2. In the *Get Refresh Token* window, select *Authorization Code*. 3. Enter an authorization code. To get an authorization code, follow the instructions in the Google documentation (https://developers.google.com/identity/protocols/oauth2/native-app#step-2:-send-a-request-to-googles-oauth-2.0-server). 4. Click *OK*. A new refresh token is generated and appears (masked for privacy and security reasons) in the *Refresh Token* field .
Web Service URL	Specifies the Data Services Web Services server URL that the data flow uses to access the Web server.
Compression Type	Specifies the type of compression to use. - *None*: Does not use compression. - *gzip*: Enables you to upload gzip files to GCS. > **Note** > If you set the *Batch size (rows)* option to a nonzero positive integer, Data Services ignores the *Compression Type* of *gzip*. *Batch size (rows)* is in the target file format editor. Applicable only for flat files, XML files, and JSON files.
Connection Retry Count	Specifies the number of times Data Services tries to create a connection with the remote server after the initial connection attempt fails. The default value is **10**. The value cannot be zero. After the specified number of retries, Data Services issues an error notification and stops the job.

Option	Description
Batch size for uploading data, MB	Specifies the maximum size for a block of data to upload per request. The limit is 5 TB. For each request, Data Services uploads a block of data in the specified size from the local temporary file to your GCS account.
Batch size for downloading data, MB	Specifies the maximum size for a block of data to download per request. The limit is 5 TB. For each request, Data Services downloads a block of data in the specified size from GCS to your local file.
Number of threads	Specifies the number of threads to run in parallel when transferring data to GCS. Enter a number from 1 through 30. The default is 1. Processing with multiple threads can increase job performance. If you enter any number outside of 1 through 30, Data Services automatically adjusts the number at runtime. For example, Data Services changes a setting of 50 to 30 at runtime.
Bucket	Specifies the GCS bucket name, which is the name of the basic container that holds your data in GCS. Select a bucket name from the dropdown list. When you upload data, you can create a new bucket by entering a name that doesn't exist in your Google account. When Google does not find a bucket by that name, it creates the bucket when you perform the upload. **i Note** If you attempt to download a bucket that doesn't exist in your Google account, Data Services issues an error.

Option	Description
Remote Path Prefix	Specifies the location of the Google Cloud Storage bucket. To enter a folder structure, ensure that the path prefix ends with a forward slash (/). For example, `test_folder1/folder2/`. You must have permission to this directory. If you leave this option blank, Data Services assumes that your GCS bucket is in the home directory of your file transfer protocol. **Source:** If source is an associated file format, Data Services accesses the remote directory and transfers a copy of the data file to the local directory for processing. **Target:** If target is an associated file format, Data Services accesses the local directory and transfers a copy of the generated file to the remote directory.
Local Directory	Specifies 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. **Note** If this option is blank, Data Services assumes the directory `%DS_COMMON_DIR%/workspace` as the default directory. **Source:** If source is an associated file format, Data Services accesses the remote directory and transfers a copy of the data file to the local directory for processing. **Target:** If target is an associated file format, Data Services accesses the local directory and transfers a copy of the generated file to the remote directory.
Proxy Host, Port, User Name, Password	Optional. Species the proxy server information when you use a proxy server.

Parent topic: Google Cloud Storage file location [page 112]

Related Information

Decrease local disk usage when uploading [page 119]

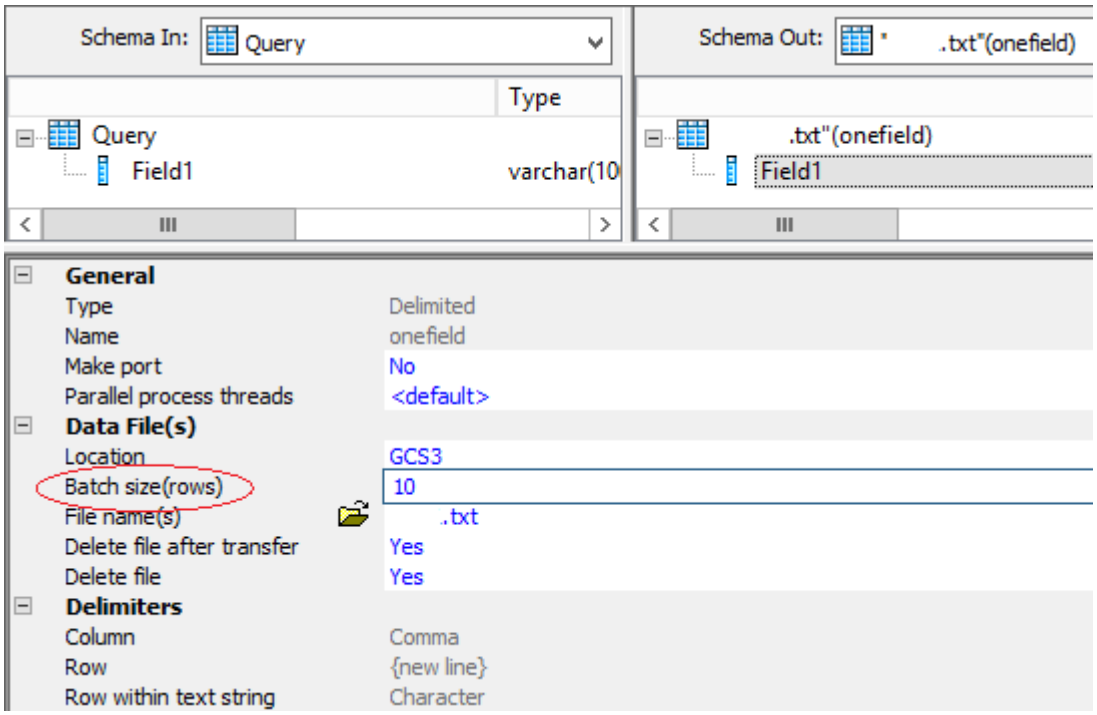
4.2.4.2 Decrease local disk usage when uploading

Decrease the local disk space used when SAP Data Services uploads a large file of generated data to your Google Cloud Storage (GCS) account.

Data Services uses the file transfer protocol in a GCS file location to load large amounts of data to your GCS account. Large size is 10 MB or more. During the load process, Data Services first loads all records to a local temporary file before it uploads data to your GCS account. The first load process uses local disk space equal to the size of the file.

Decrease local disk space usage by using the option *Batch size (rows)*. The option limits the load to a set number of rows. Data Services loads the set number of rows in batches to the local file, which decreases the local disk space used.

Find the *Batch size (rows)* option in the target editor for flat files, XML files, and JSON files. The following figure shows the setting in the target editor for a flat file:



When you set *Batch size (rows)* to 1 or more, Data Services uploads the batch rows in a single thread. Therefore, Data Services ignores the setting in *Parallel process threads*. Additionally, Data Services does not use compression, so it ignores a setting of *gzip* for *Compression type* in the GCS file location.

> ⚠ Caution
>
> If you set *Location* or *File location* to a location and file name for a non-GCS file location object, Data Services ignores the setting in *Batch size (rows)*.

Disable *Batch size (rows)* by leaving the default setting of zero (0).

> **Note**
>
> The following example uses simple row and file sizes for illustration. Sizes may not realistically reflect actual data.

> **Example**
>
> Your generated file contains 10,000 rows and is about 10 MB. You set *Batch size (rows)* to 2000, which is 2 MB. Then you execute the job.
>
> Data Services loads the first 2000 rows to the local temporary file, using 2 MB of local disk space. After all 2000 rows finish loading, it uploads the 2000 rows from the local file to your GCS account. When it finishes loading the 2000 rows to GCS, Data Services deletes the rows in the local file, which frees up 2 MB of local disk storage. Data Services performs this process five times for a total of 10,000 rows. The maximum amount of local disk storage the process uses is 2 MB, compared to 10 MB without using the *Batch size (rows)* option.

Parent topic: Google Cloud Storage file location [page 112]

Related Information

GCS file location option descriptions [page 113]

4.2.4.2.1 Row batch size considerations

Consider your environment and your data before you set the *Batch size (rows)* option.

Before you set the option *Batch size (rows)* in the target file format editor, consider your goals. Also keep in mind the size of one row of data. When each row contains many fields, the record size may determine the number of rows to set.

The following table describes scenarios for three specific goals.

Goal and action scenarios

Goal	Action
You want the best performance regardless of the local disk space the process uses.	Set *Batch size (rows)* to zero (0) to disable it, and set the performance optimization settings in the GCS file location. You achieve the best performance when you use the settings in the file location object. However, the upload uses local disk space equal to the size of the entire file.

Goal	Action
You want to conserve local disk space, but you're also interested in optimizing performance.	Use *Batch size (rows)*. Run test jobs to determine the setting that best meets your goal. • A higher setting uses more local disk space, but still less space than without using the option. • A lower setting uses less local disk space than without the option.
You don't want to use more than a specific amount of local disk space.	Work backwards to determine the setting for *Batch size (rows)*, beginning with the maximum amount of disk space to use for the upload transaction.

Examples

Example

Best performance

You plan to execute the job that uploads data to your GCS account over night when there isn't a large demand for local disk space. Therefore, you leave the *Batch size (rows)* option set to the default of zero (0). Then you configure the GCS file location and set the following options for optimized performance:

- *Compression Type*
- *Batch size for uploading data, MB*
- *Number of threads*

> **Note**
>
> The settings in the GCS file location do not affect loading the data to the local temporary file, but they affect loading from the temporary file to GCS.

Example

Compare settings

The following table compares option settings for *Batch size (rows)* and how it affects local disk storage and performance. For this example, generated data contains 10,000 rows and the total size is 10 MB.

Compare settings

Settings	Example 1	Example 2
Batch size (rows)	5,000	1,000
Total disk space	5 MB	1 MB
Number of connections	2	10

Settings	Example 1	Example 2
Performance (up ↑ or down ↓)	↑	↓

Example

Maximum batch size

You design the Data Services data flow to add additional columns to each row of data, making each row longer and larger than before processing. You base the number of rows to set for *Batch size (rows)* using a maximum disk space.

- Maximum disk space to use is 2 MB
- Convert MB to Bytes: 2 MB = 2,097,152 Bytes
- Generated row size is 1,398 bytes
- 2,097,152 ÷ 1,398 = 1,500
- Set *Batch size (rows)* to 1500

If the generated data contains 10,000 rows, Data Services connects to GCS a total of seven times, with the last batch containing only 500 rows.

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.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

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.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

THE BEST RUN SAP