



Configuration Guide | PUBLIC

Document Version: 1.19 – 2025-01-17

SAP Identity Management Configuration Guide

Content

1	SAP Identity Management Configuration Guide.	10
2	Introduction.	11
2.1	Architecture Overview.	11
2.2	Typical Implementation Workflow.	13
2.3	Factors that Influence Performance.	15
2.4	Use Cases.	19
	Prerequisites.	20
	SuccessFactors Employee Central Integration.	22
	SAP HCM Integration.	23
	SAP NetWeaver Portal Environment.	25
	Identity Lifecycle Management.	27
	Enhanced SAP Business Suite Integration.	28
	Integrating a Central User Administration System.	32
2.5	Concepts.	33
	Access Control.	33
	Approval Processing.	34
	Attestation.	41
	Attribute Ownership.	47
	Attribute Validity.	48
	Attribute DateTime Handling.	53
	Audit Log.	55
	Assignment Grouping.	57
	Context Based Assignments.	60
	Context Variables.	62
	Dispatchers.	67
	Encryption.	68
	Entry Types.	71
	Forms.	71
	Group Concepts.	72
	Identity Store.	74
	Identity Store Attributes.	76
	Identity Management User Interface.	76
	Inactive Entries.	77
	Jobs and Passes.	78
	Member Event Handling.	81
	Obsolescence.	87

	Parallel Provisioning.	89
	Packages.	91
	Package Constants.	93
	Pending Value Objects.	95
	Performance Optimization.	95
	Privileges.	97
	Provisioning.	98
	Reconciliation.	99
	Repository Types, Constants and Variables.	101
	Repository Specific Job Exection.	105
	RequestID.	105
	Statement Execution Analysis.	107
	Tracing.	107
3	Configuring SAP Identity Management.	108
3.1	SAP Identity Management Developer Studio.	108
	Logging on to the Identity Management Developer Studio.	109
	Switching Users in the Identity Management Developer Studio.	110
	Authentication and Authorization	111
	Maintaining Identity Center Properties.	113
	Maintaining Identity Stores.	124
	Maintaining the Identity Store Schema.	165
	Maintaining Packages.	189
	Maintaining Repository Types.	211
	Maintaining Processes.	218
	Maintaining Forms.	270
	Maintaining Jobs.	289
	Searching for an Object in the Configuration.	394
	Locating Parent Objects in the Tree View.	395
	Closing Related Views for an Object.	396
	Changing the Display Language of Attributes, Entry Types and Forms.	396
3.2	Provisioning Framework for SAP Identity Management 8.0.	399
	Package Files.	399
	Package Types.	404
	Package Dependencies.	404
	Package Content.	405
	Importing the Provisioning Framework for SAP Identity Management 8.0.	487
	Updating the Provisioning Framework.	491
3.3	Maintaining Repositories.	492
	Creating Repositories.	493
	Enabling Repositories.	495
	Disabling Repositories.	496

	Copying Repositories.	497
	Exporting Repositories.	497
	Importing Repositories.	498
	Deleting Repositories.	499
	Changing Repository Type of Repositories.	499
3.4	Maintaining Identity Store Entries.	501
	Maintaining Identities.	502
	Maintaining Roles.	506
	Maintaining Privileges.	522
	Maintaining Dynamic Groups.	540
	Handling Basic and Assignment Approvals.	547
	Viewing Task History.	550
	Viewing Attribute History.	551
	Searching for Identity Store Entries.	552
3.5	Self-Service Password Reset.	553
	Creating UME Role idm.anonymous.	555
	Creating Password Reset Form.	562
	Creating Password Reset Failed Process.	565
	Configuring Password Reset Parameters.	569
	Editing Authentication Information.	573
	Testing Self-Service Password Reset.	578
	Changing Authentication Questions.	583
	Resetting Number of Failed Password Reset Attempts.	587
3.6	Configuring E-Mail Notifications.	592
	Notifications from SAP Provisioning Framework Processes.	592
	Notifications from Approval or Attestation Tasks.	593
4	Connecting SAP Systems.	594
4.1	Determine the System Landscape.	594
4.2	Setting up the Landscape.	596
	Attributes Using Value Help Content.	597
	Reading Value Help Content.	598
	Setting Up an SAP HCM System.	602
	Setting Up an AS ABAP System.	623
	Setting Up an AS Java System.	628
	Setting Up an SAP HANA System.	632
	Setting Up a SuccessFactors System.	640
	Setting up an Identity Authentication System.	645
	Setting up a SCIM System.	646
	Setting Up Hybrid Integration with Cloud System.	647
	Setting Up an S/4HANA On-Premise System.	650
	Password Provisioning.	660

	Single Sign-On for AS ABAP Systems.	662
	Preparing the Initial Loads.	663
	Running the Initial Loads.	667
	Cleaning up the Collected Data.	668
4.3	Assigning the System Account Privilege to Users or Company Addresses.	668
4.4	Using an Update Job for Reading New Privileges and New Company Addresses.	668
	Creating an Update Job.	669
	Scheduling an Update Job.	670
4.5	Assigning Function Set Privileges to Users for SAP Business Suite Applications.	671
4.6	Setting up an SAP Java Connector (SAP JCo) and Related Traces.	671
	Restricting the CPIC or JRFC Trace to a Specific Pass.	676
5	Connecting Non-SAP Systems.	679
5.1	Setting Up a Microsoft Active Directory and Microsoft Exchange.	679
	Creating and Mail-Enabling Users in Microsoft Exchange.	680
5.2	Setting Up a Lotus Notes System.	682
	Importing the Lotus Notes Schema.	684
	Importing Lotus Notes Connector Package.	686
	Adding NCSO.jar in Identity Management Dispatcher Classpath.	687
	Adjusting Attribute Values.	688
	Setting Up the Replication Source for the Mail File	689
	Configuring Password Reset.	690
	Modifying Notes Certifiers.	691
	Recertifying Notes Users.	691
	Adding New Attributes.	692
	Migrating to Java-Based Lotus Notes Connector.	693
6	Integration with SAP Access Control 10.0 or higher.	695
6.1	Integration Scenarios.	696
	Landscape Configuration Scenarios.	697
	Result Handling Scenarios.	700
	Provisioning Initiation Scenarios.	702
	Integration Scenario Combinations.	705
6.2	Business Role Synchronization.	706
	Configuring Business Role Synchronization.	707
6.3	Configuring the Virtual Directory Server.	710
	Adding the Virtual Directory Server Configuration.	711
	Modifying the (Template) Constants.	715
	Setting the Access Credentials.	718
	Starting the Virtual Directory Server.	720
	Testing the Virtual Directory Server Configuration.	720
6.4	Importing the GRC Provisioning Framework.	724

	GRC Provisioning Framework Package.	725
6.5	Configuring the Solution.	728
	Configuring GRC Initiated Provisioning.	729
	Configuring Identity Management Initiated Provisioning.	730
	Configuring Centralized Provisioning Scenario.	731
	Configuring Distributed Provisioning Scenario.	736
	Configuring the Identity Services (for Callback Service).	737
	Configuring the Callback Service Result Handling Scenario in GRC System/SAP Access Control.	739
	Grouping of Privileges.	744
	Configuring the Properties on Provisioning Framework Processes.	749
	Configuration Summary.	749
6.6	Running the Initial Load Service Jobs.	750
	AC 10.0 – Initial Load – Commons.	751
	AC 10.0 – Initial Load – Centralized Provisioning.	753
	AC 10.0 – Initial Load – Distributed Provisioning.	754
	Troubleshooting.	755
6.7	Timeouts and Stale Requests.	756
6.8	Running the GRC Provisioning Framework.	756
6.9	Updating the GRC Provisioning Framework.	756
6.10	Privilege Assignment Process.	757
	Centralized Provisioning Scenario.	757
	Distributed Provisioning Scenario.	758
	Obtaining the Process Values.	759
6.11	Business Role Assignment Process.	760
6.12	Process Execution Description.	760
	AC Validation.	761
	AC Polling.	769
	AC Callback Service.	771
	AC Validation - Risk Analysis Only.	772
6.13	GRC Related Repository Constants.	774
6.14	Limitations/Tuning of the Framework.	778
7	Configuring Reports.	780
7.1	Standard Reports.	780
	Executing Reports.	783
	Creating or Modifying Reports.	784
7.2	Identity Reporting Using SAP Business Warehouse.	787
	Overview of the System Landscape.	788
	Overview of the Implementation Steps.	792
	Setting up Reporting with SAP BW.	794
	Using the Web Templates.	808

	Additional Tasks.	810
8	SAP Identity Management REST Interface Version 2.	828
8.1	Security.	829
	Authentication and Authorization.	829
	Cross-Site Request Forgery Protection.	830
8.2	Content Type Negotiation.	831
	Content Type Via Request Header.	831
	Content Type Via System Query Option.	832
	Character Encoding.	832
8.3	Localization.	833
8.4	Performance.	834
	OData Means.	834
	Response Properties.	835
8.5	Comparison to the Previous Version of the Identity Management REST Interface.	835
	URIs.	836
	Operations.	836
	Return Format.	837
	Query Options.	838
8.6	Identity Store Support.	839
8.7	Limitations.	839
8.8	Compatibility.	839
8.9	Configuring the UME Actions for the REST Interface.	842
8.10	Configuring Single Sign-On With Logon Tickets in the REST Interface for AS Java 7.1 and higher	842
8.11	General Service URIs.	844
	Service Document.	844
	Service Metadata Document.	845
8.12	Entry Access URIs.	846
	Access to Security Relevant Attributes.	847
	Support of Binary Attributes.	848
	Create Entries.	848
	Search Entries.	851
	Get Specific Entry.	857
	Update an Entry.	863
	Get Attribute Values of an Entry.	870
	Add Attribute Values to an Entry.	878
	Update Attribute Values of an Entry.	882
	Search Entry References.	884
	Get Entry Reference History.	891
	Get Entry Reference Hierarchy.	893
	Context Usage.	896

	Virus Scanning.	896
8.13	Entry Type Access URIs.	897
	Get All Entry Types.	898
	Get a Specific Entry Type.	903
8.14	Task Access URIs.	909
	Get All Self-Service Tasks.	909
	Get a Specific Self-Service Task.	912
	Get All Task Attributes.	915
	Get a Specific Task Attribute.	918
	Get Allowed Values of a Specific Task Attribute.	922
8.15	Approval Access URIs.	926
	Get Open Approval Requests for Current User.	926
	Process an Approval.	929
8.16	Attestation Access URIs.	937
	Get Attestation Requests for Current User.	938
	Process an Attestation.	942
	Get a History of Attestations.	946
8.17	Service Operations.	949
	Get Version Information.	949
	Get Valid Decision Options for Approval.	951
	Execute Approval Decision.	952
	Get Valid Decision Options for Attestation.	959
	Execute Attestation Decision.	961
	GRC Notification.	963
9	Reference.	965
9.1	Reference.	965
	Identity Center Properties.	965
	Identity Store Properties.	971
	Dispatcher Properties.	974
	Entry Type Properties.	981
	Attribute Properties.	987
	Package Folder Overview.	1003
	Package Properties.	1003
	Constants (Package).	1011
	Repository Type Properties.	1012
	Job Properties.	1022
	Repository Job Properties.	1027
	Pass Properties.	1029
	Constants (Job).	1083
	Variables (Job).	1085
	Process Properties.	1086

	Form Properties.	1091
	Task Properties.	1109
	Internal Functions.	1130
9.2	Appendix.	1234
	Repository Constants.	1234
	Attribute Mappings.	1276
	Attributes that Support Value Help.	1311
	Configuring the ABAP Connector to Use SNC.	1313
10	Restrictions.	1330
10.1	Restrictions and Considerations that Apply to Specific Use Cases.	1330
	Restrictions When Using the SAP HCM Use Case.	1330
	Restrictions When Using the SAP Enterprise Portal.	1331
	Restrictions When Using the Identity Lifecycle Management Use Case.	1331
	Restrictions When Using the Enhanced SAP Business Suite Integration Use Case.	1331
10.2	Restrictions and Considerations that Apply to Specific Connectors.	1332
	Restrictions that Apply to All Connectors.	1332
	Restrictions for AS ABAP/SAP Business Suite System Connectors.	1333
	Restrictions when Connecting a CUA System.	1334
	Restrictions for AS Java System Connectors.	1334
	Restrictions for LDAP Directory Connectors.	1335
	Restrictions for SAP HANA Connectors.	1335
	Restrictions for SuccessFactors Connector.	1336
10.3	Considerations When Customizing the Provisioning Framework.	1338

1 SAP Identity Management Configuration Guide

This document provides an overall information how to configure SAP Identity Management to meet your business needs.

Related Information

[Introduction \[page 11\]](#)

[Determine the System Landscape \[page 594\]](#)

[Configuring SAP Identity Management \[page 108\]](#)

[Connecting SAP Systems \[page 594\]](#)

[Connecting Non-SAP Systems \[page 679\]](#)

[Integration with SAP Access Control 10.0 or higher \[page 695\]](#)

[Identity Reporting Using SAP Business Warehouse \[page 787\]](#)

[SAP Identity Management REST Interface Version 2 \[page 828\]](#)

[Reference \[page 965\]](#)

[Restrictions \[page 1330\]](#)

2 Introduction

This section provides the following information:

- Architectural overview over system landscapes when using the SAP Provisioning Framework
- Implementation workflow of the identity provisioning in SAP Identity Management
- Factors that influence the performance of SAP Identity Management
- Overview of use cases and how you can use them to provision identity data in your landscape.
- Key concepts when using SAP Identity Management.

Related Information

[Architecture Overview \[page 11\]](#)

[Typical Implementation Workflow \[page 13\]](#)

[Factors that Influence Performance \[page 15\]](#)

[Use Cases \[page 19\]](#)

2.1 Architecture Overview

Enterprises are under pressure to increase the speed of deploying new applications and systems across their global networks, both internally and in the context of e-business with partners and customers. One of the challenges involved in these processes is the difficulty in finding and bringing together information relating to identities and resources that are distributed across multiple and often incompatible information sources.

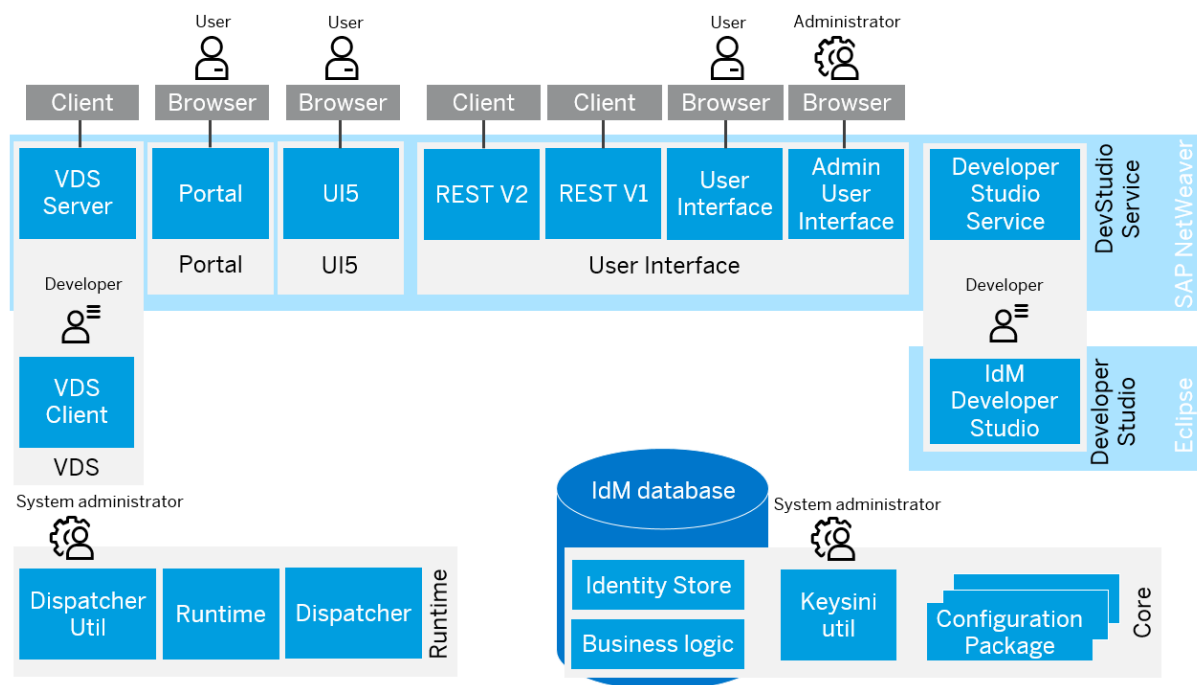
The prime objective of identity management is to centrally manage and keep all identity data within the enterprise up-to-date. Identity data is often stored in many different applications throughout the enterprise and maintained manually in different locations. This is costly and, in addition to posing a security risk, can cause inconsistencies and low data quality.

SAP Identity Management provides the functions and services needed to integrate distributed identity data in the system landscape to efficient, heterogeneous identity lifecycle management.

Architecture

SAP Identity Management consists of several components. Some of the components run on SAP NetWeaver Application Server Java, for example, the Identity Management User Interface. Other components are stand-

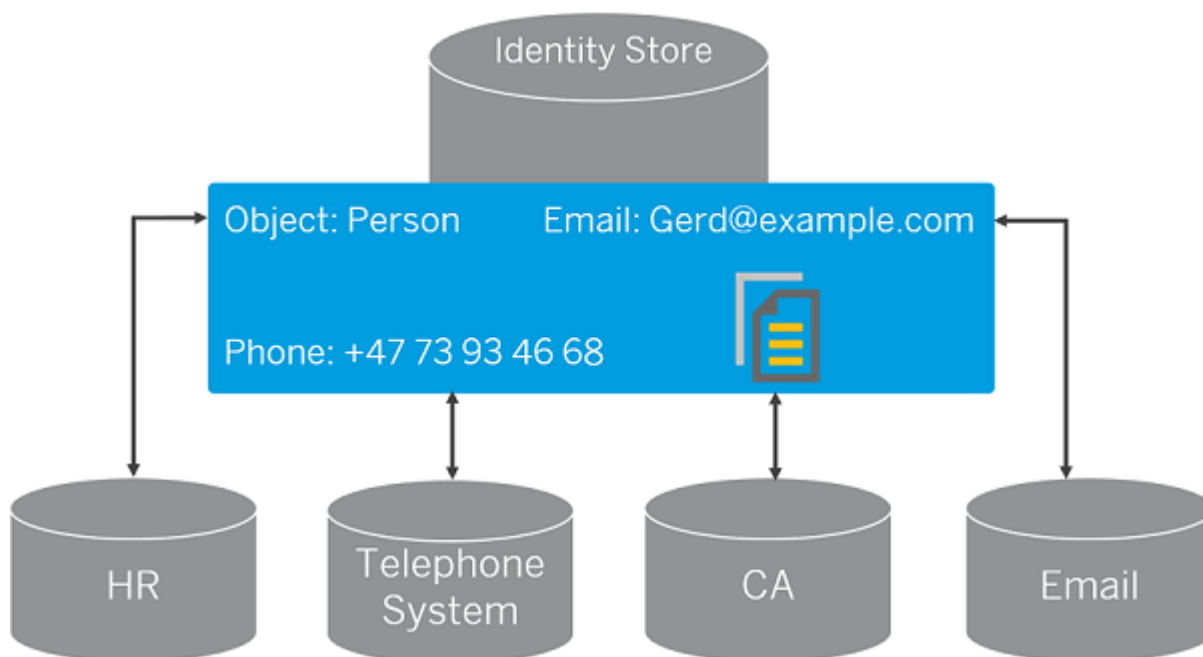
alone and are installed separately. The complete set of software units that make up SAP Identity Management are categorized as follows:



For more information, see *Software Components of SAP Identity Management* in *SAP Identity Management Master Guide*.

Identity Center

The Identity Center includes functions for identity provisioning, workflow, password management, logging, and reporting. It uses a centralized repository, called the identity store, to provide a uniformed view of the data, regardless of the data's original source. This architecture is shown in the figure below.



The Identity Center retrieves the data from these various repositories, consolidates it, transforms it into the necessary formats, and publishes it back to the various decentralized repositories.

📌 Note

In SAP Identity Management 8.0 documentation, the Identity Center is used to designate the following parts it consists of: Identity Management database, Runtime components and Identity Management Developer Studio.

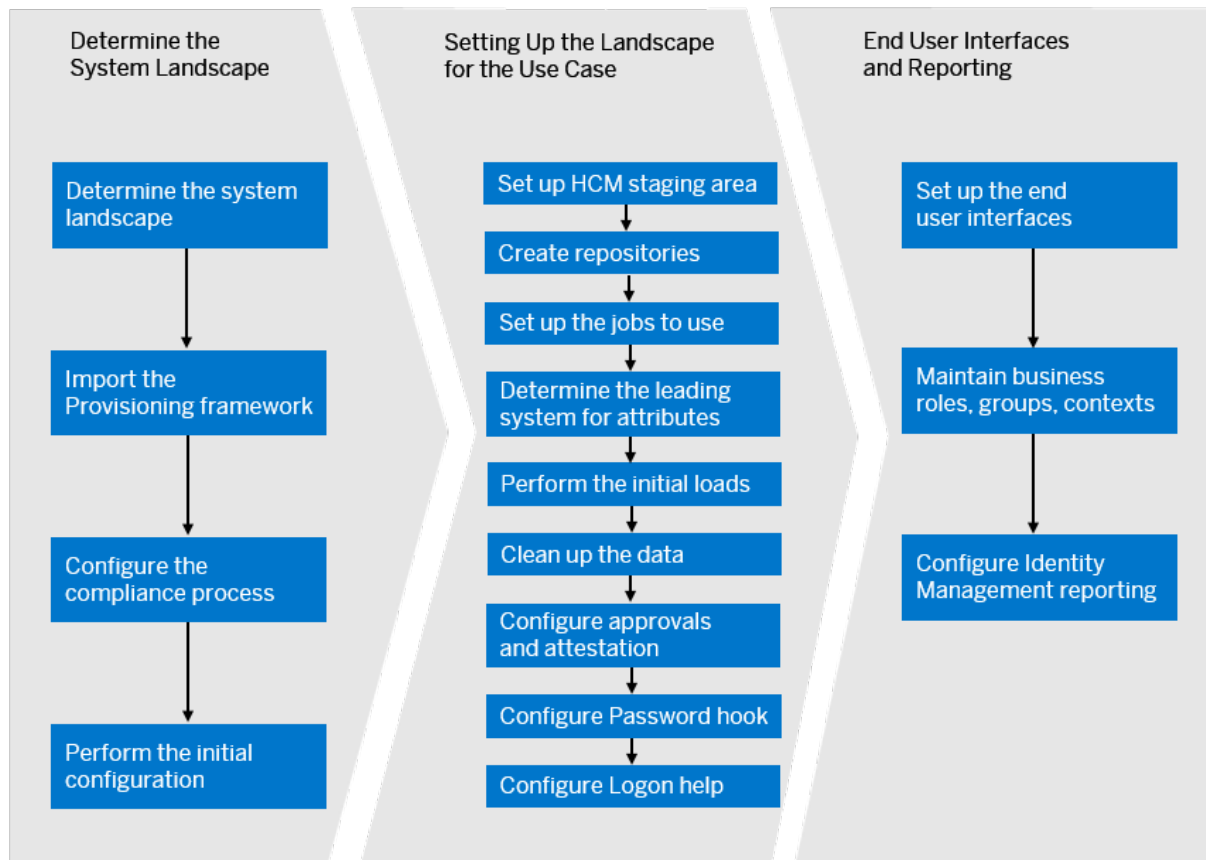
Related Information

[Software Components of SAP Identity Management](#)

2.2 Typical Implementation Workflow

Context

The implementation workflow of the identity provisioning in SAP Identity Management based on the templates we provide, involves the following steps:



Procedure

1. Determine the system landscape. Identify your leading identity system and your target systems. For more information, see [Determine the System Landscape \[page 594\]](#)
2. Import the provisioning framework for SAP systems into the SAP Identity Management Identity Center choosing the necessary packages. For more information, see [Importing the Provisioning Framework for SAP Identity Management 8.0 \[page 487\]](#)
3. Configure compliance process.
4. Perform the initial configuration.
5. Set up the landscape for the use case. For more information, see [Setting up the Landscape \[page 596\]](#)
 - For system landscapes where you use the SAP HCM system as the leading system for employee data, you need to set up a staging area in the Identity Center, set up the Virtual Directory Server, configure the SAP HCM system and maintain the attribute mappings.
 - Creating repositories for each system that you connect to the Identity Center. For more information, see [Creating Repositories \[page 493\]](#)
 - Set up the jobs to use for the use case.
 - Determine the leading system for attributes that use value help and read the value help from this system.

- Import the identity data into the Identity Center's identity store by performing the initial loads.
 - Clean up the data that was collected from the initial loads. Changes will be provisioned back to the connected systems.
 - Configure workflows: approvals and attestation
 - Configure Password hook
 - Configure Logon help
6. Set up the end user interfaces for performing user administration or create your custom UIs. You have the following options:
 - To create and configure UI tasks (forms), see [Maintaining Forms \[page 270\]](#).
 - To create your custom UIs, see [SAP Identity Management REST Interface Version 2 \[page 828\]](#) or [SAP Identity Management Identity Center REST API](#) .
 7. Maintain the identity store entries, such as business roles, privileges, groups, dynamic groups in the Identity Management User Interface. For more information, see [Maintaining Identity Store Entries \[page 501\]](#)
 8. Configure Identity Management reporting. You have the following options:
 - Use [Standard Reports \[page 780\]](#) delivered as predefined forms in SAP Identity Management User Interface.
 - Use SAP Business Warehouse (BW) for reporting on identities managed by SAP Identity Management. For more information, see [Identity Reporting Using SAP Business Warehouse \[page 787\]](#)
 - Create your own reports. For more information, see [SAP Identity Management HTML Reporting: How to create HTML Reports](#) .
 - Create reports for Identity Management data in SAP Lumira. For more information, see [Identity Management reporting via SAP Lumira](#) .

2.3 Factors that Influence Performance

There are several factors that influence the performance of SAP Identity Management. The list is by no means exhaustive. These are some of the factors that you should consider when configuring and running your SAP Identity Management system.

If you still experience performance issues after following all recommendations and considerations of this section, see [How to troubleshoot performance issue of an SAP Identity Management system](#) .

Dispatchers

You should configure only one running dispatcher in your system. Deadlock may happen when multiple dispatchers are simultaneously executing the same action, and this will affect system performance negatively. For more information about how many dispatchers should be configured for an Identity Management system, see SAP KBA [2437631](#) .

Which configuration gives the most optimal performance of the dispatcher will vary depending on a number of factors and must be based on experience with the specific system. For more information, see [Configuring the Dispatcher Policy](#)

SQL Queries as Part of the Configuration

- If you need to read a large dataset from Microsoft SQL Server, SAP ASE and IBM DB2, then it is recommended to use the relevant mechanisms for reading uncommitted data in all SELECT queries (including their subqueries). This will reduce the table locking, improve performance, and reduce the probability of deadlocks. For example:
 - In SAP ASE - `AT ISOLATION READ UNCOMMITTED`
 - In IBM DB2 - `WITH UR`
 - In MS SQL - `WITH (NOLOCK)`

Note

Oracle database does not support a direct mechanism for reading uncommitted data.

For more information refer to the official documentation of the database in use.

Storing such data to the Identity Management database involves making numerous updates. Therefore, if the read from the database takes X time, it is not the case that the update will also take the same time. The same applies for other databases.

Although using mechanisms for reading uncommitted data can improve the query performance, be aware that it can also introduce the possibility of dirty reads (read of uncommitted data).

- Use the indexed column `searchvalue` instead of `value` in SQL queries. See [Analyzing Statement Execution](#)
- When using dynamic groups, consider carefully the performance of any SQL statements used to resolve the group members. Extensive use of dynamic groups is not recommended due to the possible impact on performance. Furthermore, when defining a filter for a dynamic group, note that using `MSKEYVALUE`, `DISPLAYNAME` or `MX_ENTRYTYPE` attributes in combination with other attributes will impact the performance of the SQL query execution.

Rebuilding Database Indexes

With heavy usage of the system, the database indexes will become fragmented, which may decrease performance. For more information about fragmented indexes and rebuilding the indexes, refer to the documentation for your database system.

Most often, if your system is suffering this issue, you will see system wide performance issues. The user interface (UI), jobs and tasks will all perform more slowly than normal, or the system could, in most severe cases, come to a standstill. Keeping the indexes refreshed is essential, as much of the processing in the Identity Management system occurs at database level.

Job Execution

- **Long running jobs**

Running an initial load job from an ABAP system that has many ABAP roles and profiles may take longer than expected. It can take some time to read all these into the Identity Management and this action can timeout. You can increase the timeout on the [General](#) tab of the Identity Center properties. There is no performance impact on increasing the timeout values, but if for example network problems prevent the jobs from running normally, it will take longer before a timeout error is detected and logged. See [1352027](#) 🗉

- **Heap size**

Increasing the heap size in the dispatcher prop file has a performance impact. However, you may need to set the maximum heap size in the dispatcher prop file when getting `OutOfMemoryError` due to reading large amount of data (for example, from an ABAP system or UME). In this case, we do not recommend setting the maximum heap for the main dispatchers. Instead, create a separate dispatcher for running large jobs, and do this change only for this dispatcher.

UI Configurations

- Opening a form in the UI may take longer if there is a large number of values to load. This is relevant for forms with reference attributes assigned to them. To speed up the loading of the form, you can disable the [List Entries on Load](#) option on the [Presentation](#) tab of the attributes. Thus, the attributes will no longer be loaded automatically in the UI, and the user must search for them after the UI opens.
Opening a form in the UI may also take longer if attributes with [SingleSelect](#) presentation type have a large list of predefined values. To speed up the loading of the form, make sure you do not have a large list of values.
- Complex access controls on forms can also affect the performance in the UI. Check the SQL statement used in the access control and see if it can be refined to make it faster.
- Using access limitations to restrict a user to display or search specific entries impacts performance. This is due to the following reason:

When searching for an entry type which has applied access limitations, the access limitations must be checked for each entry that has applied access limitation before searching or displaying it. For example, when searching for roles and displaying role memberships, the access limitations must be checked for every role before displaying it.

⚠ Caution

If the value of the attribute that is defined as a search attribute and a user attribute contains asterisk (*), this significantly slows down the search process. We recommend that you avoid using asterisk (*). Instead, define the attribute as a multivalue attribute and add the corresponding values. In this case, the user's user attribute is compared to the entries' search attribute, and only those entries that have the same value as the user's user attribute are displayed.

- Evaluating the access control of the favorite tasks may have impact on performance. This should be considered before enabling the [Show favorite task bar in display tasks](#) option on the [General](#) tab of the identity store.
- Enabling the following options: [Show Future Values](#), [Show Assignment History](#), [Show All Values](#) and [Show Historic Values](#) on the [Presentation](#) tab of a form will make the form to retrieve more data from the

database and will affect performance. The retrieved data should be for administrative purpose instead of all end users.

Provisioning Process

- If you have a large number of repositories of the same type, and many entries to be processed to several repositories at the same time, consider using parallel provisioning which improves performance by allowing provisioning to different repositories to run in parallel. See [Parallel Provisioning](#)
- Setting up identity center properties and provisioning constants can also improve performance when it comes to modify events on privileges and direct privilege inheritance. For more information, see [Performance Optimization](#)
- Use the [Provisioning Monitor](#) tool to check provisioning queue issues.
- When importing the data from SAP HCM system into the staging area in the Identity Center, note that the delta mechanism is not preconfigured. A full load is always performed. For more information about delta processing during SAP HCM integration with SAP Identity Management, see [1910756](#) . For performance problems with export from SAP HCM, see [1711869](#) 

Event Processes

Adding event processes for an entry type can affect performance and should only be used if strictly necessary.

Reconciliation

Performing structural changes to role hierarchy triggers the reconciliation process. This may affect multiple entries and therefore be very time consuming and process intensive. One such example is a change to a role hierarchy with a large number of roles and thousands of users, which all are assigned inherited roles. If one of the branches in the role hierarchy is added or removed, the roles and privileges of all the thousands of users must be recalculated, which may put a very high load on the system for a long time. See [Reconciliation](#)

Be aware that the performance of reconciliation depends on:

- Complexity of the role hierarchy. Try to use simple role hierarchy and try to avoid multiplied child business roles or privileges - appear more than once at different hierarchy levels or different child business roles.
- Number of affected users
- Your system load at that time

Updating Attributes

It is strongly recommended to use a To identity store pass for updating attributes in the identity store. For performance reasons and to avoid deadlocks in the database, no entry event processes are started, and it is

not verified that the attribute is unique. Updating attributes in this way may lead to duplicates in the identity store.

REST Interface Version 2

For more information about factors that influence REST Interface Version 2 performance, see [Performance](#)

2.4 Use Cases

You can use SAP Identity Management for processing identity information in a variety of ways, depending on your system landscape. You can use it in homogeneous or heterogeneous landscapes, either with or without SAP systems.

This document shows a few sample use cases where you can use SAP Identity Management for identity provisioning with SAP systems. It is not meant as a complete covering of all options; however, you should be able to adapt the information provided as guidelines when setting up SAP Identity Management in your own system landscape.

The use cases covered by this document are:

- **SuccessFactors Employee Central (EC) Integration**
This use case shows how to manage identities when the leading identity source is SuccessFactors Employee Central and the identities are provisioned to an LDAP directory server, an AS Java system that runs an SAP Enterprise Portal, an AS ABAP system, and back to SuccessFactors.
- **SAP Human Capital Management (HCM) Integration**
This use case shows how to manage identities when the leading identity source is an SAP HCM system and the identities are provisioned to an LDAP directory server, an AS Java system running a portal, and an AS ABAP back-end system.
- **SAP Enterprise Portal Environment**
This use case shows how to manage identities in an SAP Enterprise Portal environment. In this case, the leading identity source is a corporate directory, and the identities are provisioned to the portal's AS Java and the various back-end systems. In this example, we show how to provision to an AS ABAP back-end system.
- **Identity Lifecycle Management**
This use case describes an integrated, heterogeneous landscape that includes the SAP HCM and portal use cases, as well as additional systems, for example, non-SAP systems.
- **Enhanced SAP Business Suite Integration**
Another example of a complete system landscape includes enhanced SAP Business Suite integration. In this case, additional SAP systems, which may also have special requirements, are connected to SAP Identity Management for identity provisioning. The SAP HCM system is typically the leading system for this use case.
- **Central User Administration (CUA) Integration**
As of Release 7.0 SP 2, SAP Identity Management supports the integration of a central user administration (CUA) system. In this case, you connect the CUA central system to the Identity Center as a target system.

The Identity Center provisions the identity data to the CUA central system, which in turn provisions the data to its child systems.

For each of these use cases, we describe how you can use SAP Identity Management to manage and provision users, groups, role assignments, and other identity data.

To implement these use cases, we provide a provisioning framework for SAP systems. This framework provides templates for connecting SAP systems (including SAP Business Suite systems) to SAP Identity Management and for setting up the corresponding provisioning jobs.

2.4.1 Prerequisites

The prerequisites for using the use cases include performing an initial load of the data and defining a role model.

Related Information

[Initial Load \[page 20\]](#)

[Defining a Role Model \[page 21\]](#)

2.4.1.1 Initial Load

Regardless of which use case you are focusing on, the first step that is required is to connect the various systems that contain identity data to the Identity Center and replicate the data by performing an initial load.

Note the following:

- Before reading the data into the Identity Center, you can specify which system is the source system for each attribute. For example, you can specify that the SAP HCM system is the source system for an identity's first and last name, and the mail server is the source system for the identity's e-mail address.
- When reading the data, the user ID is the attribute used for consolidating identities from the various systems. This means that a user account is created in the Identity Center's database for each unique user ID that is read from the various data sources.
- Once the data is read into the Identity Center and consolidated, provision the data back to the connected systems. This overwrites any data that should be replaced with higher-quality data.

📘 Note

The SAP HCM system is an exception to this rule. The templates provided with the SAP provisioning framework do not support provisioning to the SAP HCM system.

In the SAP HCM use case, we assume that the identity data provided by the SAP HCM system is high-quality data that does not need to be overwritten by data provided by other systems.

2.4.1.2 Defining a Role Model

You must define a role model before working with any of the use cases described in this document. The role model states which roles you have in your company and how they are reflected in the various systems.

When working with SAP Identity Management, we recommend defining the role model according to business roles and technical roles. These roles have the following characteristics:

Business Roles

- Abstract description of the role
- Company-specific semantic
- Available in the Identity Center
- Can be organized in a hierarchy with inheritance
- Examples: MANAGER or EMPLOYEE

Technical Roles or Privileges

- Technical description of the business role for a particular system.

Note

The technical roles are represented as privileges in the Identity Center.

- System-specific semantic
- Available in the originating system; replicated to the Identity Center.
- Are organized in a flat structure and inherited through the role hierarchy.
- Examples: ABAP roles, UME (user management engine) roles, portal roles, LDAP groups

In the role model, map your business roles to the corresponding technical roles for each of these systems. Set up this role model and the mapping between the business roles and the technical roles in the Identity Management so that the Identity Management can provision the role assignments correctly based on this information.

Example: Mapping a Business Role to Technical Roles

For example, the business role EMPLOYEE has a different meaning in each of the various systems.

- In the SAP HCM system, the employee can maintain certain data, for example, his or her bank account or home address.
Therefore, the business role EMPLOYEE is mapped to an ABAP role, for example, Z_HCM_EMPLOYEE_ROLE.

- In the portal system, the employee can access various services that are available to all employees. Therefore, the business role EMPLOYEE is mapped to a portal role, for example, eurole (standard user role).
- In the mail system, the employee can send or receive mails under his or her user account. Therefore, the business role EMPLOYEE is mapped to the mail server's user account privileges.

Role Maintenance

Maintain the business roles in the Identity Management User Interface. In this step, assign the corresponding technical roles to the business role.

Maintain the technical roles in each target system. Assign the user to the corresponding business role(s) in the Identity Management User Interface. The user's assignment to each technical role is then provisioned to the target systems.

Related Information

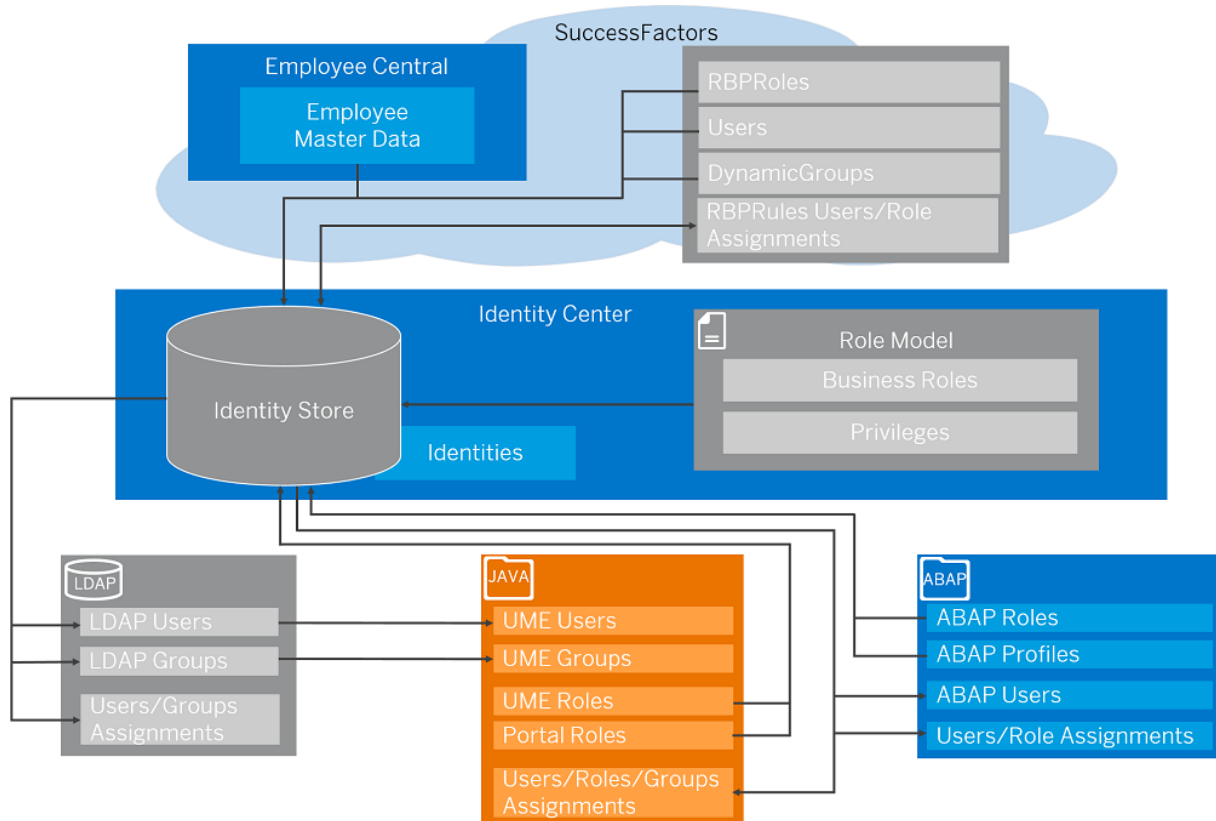
[Maintaining Roles \[page 506\]](#)

2.4.2 SuccessFactors Employee Central Integration

A scheduled SAP Identity Management initial load job gets the employee master data from SuccessFactors Employee Central provided it is available. In addition, the initial load job loads data from the user data in SuccessFactors in order to load non-employee records. Furthermore, this job supports delta load in case of changes. The initial load job also loads SuccessFactors Permission Roles and Permission Groups into IdM. The Permission Groups are represented as context entries for context-based assignments. They represent target populations in SuccessFactors role assignments (RBPRule).

The role model in the Identity Center then determines the user privileges, which are provisioned to the various target systems, covering all traditional employee scenarios: hiring a new employee, changing an employee position, employee termination and others. SuccessFactors could also be a target system for authorization assignments.

The figure below shows the use case in which the identity data retrieved from the SuccessFactors system is provisioned to an LDAP directory server, to an AS Java system that runs an SAP Enterprise Portal, to an AS ABAP system, and back to SuccessFactors.



- The Permission Role is represented as MX_PRIVILEGE.
- Employee/User is represented as MX_PERSON.
- The Permission Group is represented as MX_SFSF_TARGET.

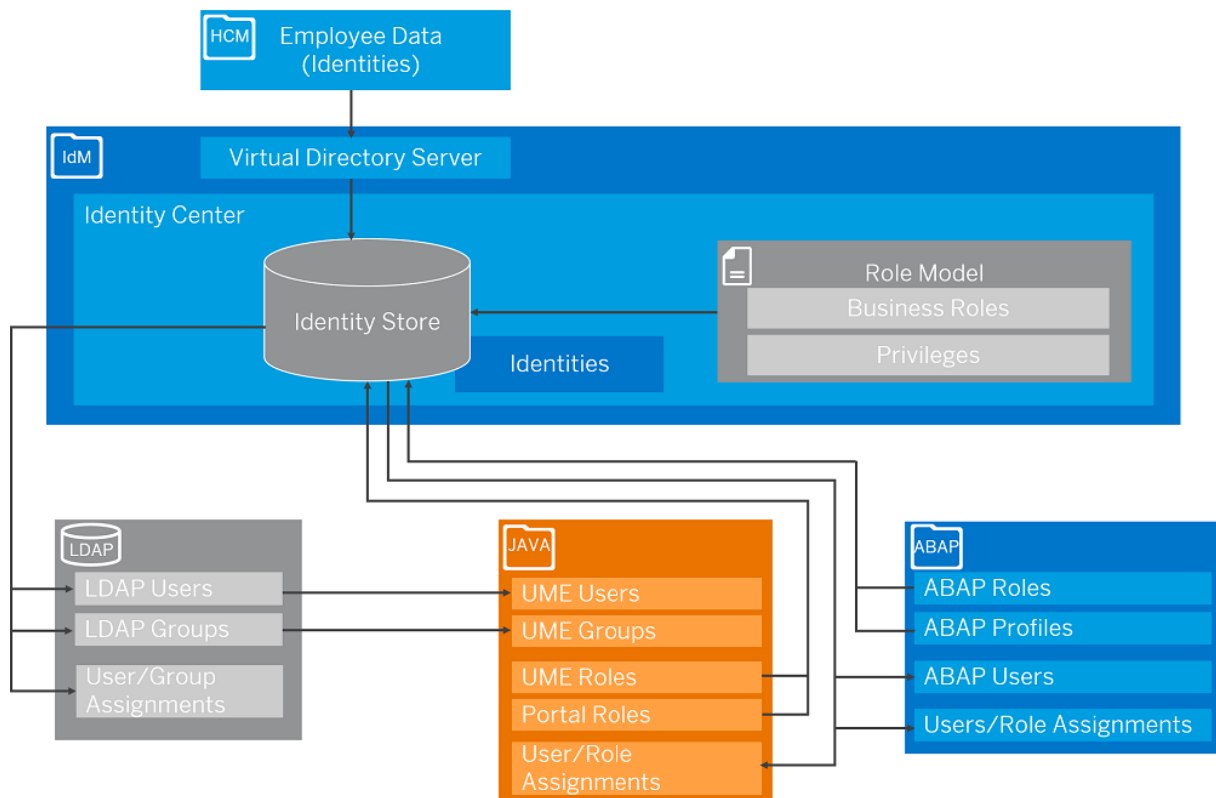
Related Information

[SAP SuccessFactors HCM Suite](#)

2.4.3 SAP HCM Integration

For this use case, identities (employee master data) are primarily maintained in the SAP HCM system. Once identities are created in SAP HCM, they are replicated to the Identity Center. The role model in the Identity Center then determines the user/role or user/group assignments that are to be provisioned to the various target systems.

The figure below shows this use case where the identity data retrieved from the SAP HCM system is provisioned to an LDAP directory server, an AS Java system that runs an SAP NetWeaver Portal, and an AS ABAP system. The AS Java system uses the LDAP directory server also as its user data source.



System Overview

A summary of the systems used in this use case are shown in the table below.

System Overview

System	Source Data	Replicated/Provisioned Data
SAP HCM	Identities (Employee Data)	-
Identity Center	Role model	Identities/Users
LDAP Directory Server	-	Users, groups and group assignments
SAP NetWeaver AS Java (with Portal)	Portal roles and UME roles	Replicated from LDAP: UME users and UME groups Provisioned from Identity Center: User/role assignments
SAP NetWeaver AS ABAP	ABAP roles and profiles	ABAP users and user/role assignments

Note

The Virtual Directory Server is used as the interface between the SAP HCM system and the Identity Center.

Notes and Recommendations

When setting up this use case in the Identity Center, take the following points into consideration:

- In this use case, the SAP HCM is the leading system for maintaining identities. Local changes made to user master records in the target systems are not reflected in the Identity Center or provisioned back to the SAP HCM system.

Note

Local changes to other attributes with a different source system, for example, email addresses where the source system is a mail server, or changes to role or role assignments can be provisioned back to the corresponding source system.

- Assign the business roles to identities in the Identity Center according to the role model. The meaning of the business roles are then mapped to the corresponding privileges and the corresponding user/role assignments are provisioned to the target systems.
- With the exception of the UME users and groups on the AS Java, the provisioning process updates users, groups, and group memberships in the target systems. The AS Java uses the LDAP directory server directly as the source for the UME users and groups.
- You can use similar concepts to set up provisioning to additional AS Java and AS ABAP systems, or to non-SAP systems.

Limitations

The templates provided with the SAP provisioning framework do not use any delta mechanism when importing the data from the SAP HCM system into the Identity Center. A full load is always performed.

When working with the identity data in the Identity Center, you can use delta mechanisms.

2.4.4 SAP NetWeaver Portal Environment

In the previous use case, we described how to set up SAP Identity Management for provisioning identities where the SAP HCM is the primary user management system. In this use case, we describe how to use SAP Identity Management in an SAP NetWeaver Portal environment where the leading system for user management is a corporate LDAP directory.

The identity information contained in the LDAP directory, which consists of users and groups, is used as the data source for the portal and this data is also replicated from the LDAP directory to the Identity Center.

In addition, to be able to derive the role assignments that are to be provisioned, the ABAP roles need to be read from the ABAP system into the Identity Center and the UME and portal roles need to be read from the AS Java where the portal runs.

You can then manage the users and their role assignments from the Identity Center and provision them to the target systems. In this example, the identities are provisioned to the portal and its back-end ABAP system.

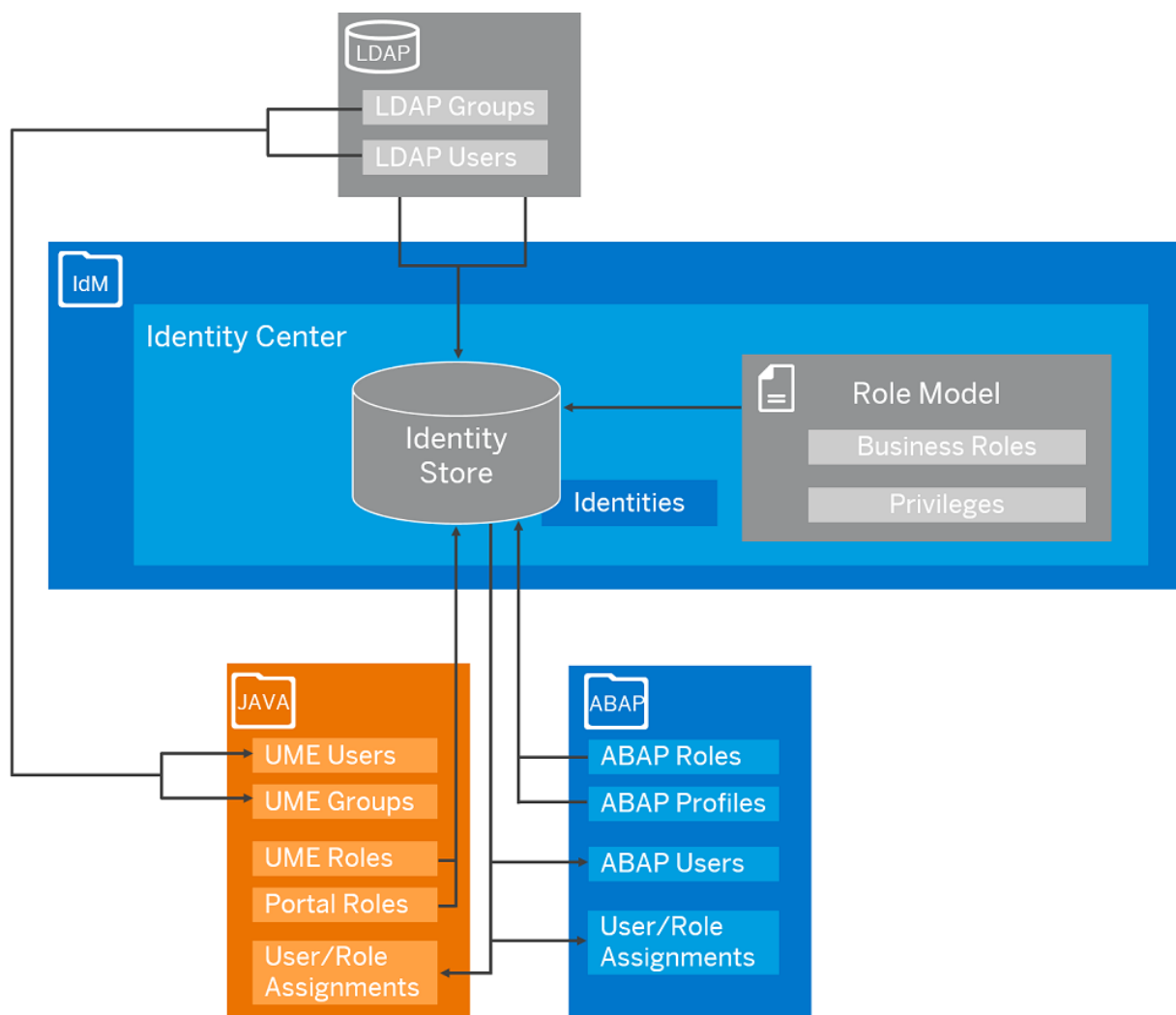
❗ Note

Since the LDAP directory server is the data source for the portal, the portal can read the user and group information directly from it, and this information does not need to be provisioned from the Identity Center. The rest of the information is provisioned to the systems accordingly.

❗ Note

For example, ABAP users and their role assignments are provisioned to the ABAP system and the role assignments for the portal are provisioned to the portal.

See the figure below.



In this way, the users who log on to the portal are provided with the correct set of portal roles together with the corresponding ABAP roles that they need to access the back-end ABAP systems.

System Overview

A summary of the systems used in this use case are shown in the table below.

System	Source Data	Replicated/Provisioned Data
LDAP Directory Server	Users and groups	-
Identity Center	Role model	Identities (Users)
SAP NetWeaver AS Java (with Portal)	Portal roles, UME roles	Replicated from LDAP: UME users and UME groups Provisioned from IC: Role assignments
SAP NetWeaver AS ABAP	ABAP roles and profiles	Users and role assignments

Notes and Recommendations

When setting up this use case in the Identity Center, take the following points into consideration:

- In this use case, the corporate LDAP directory is the leading system for maintaining identities. If you maintain user master records locally in the target system after performing the initial load into the Identity Center, these changes are not reflected in the Identity Center and are not included in the provisioning process.

Note

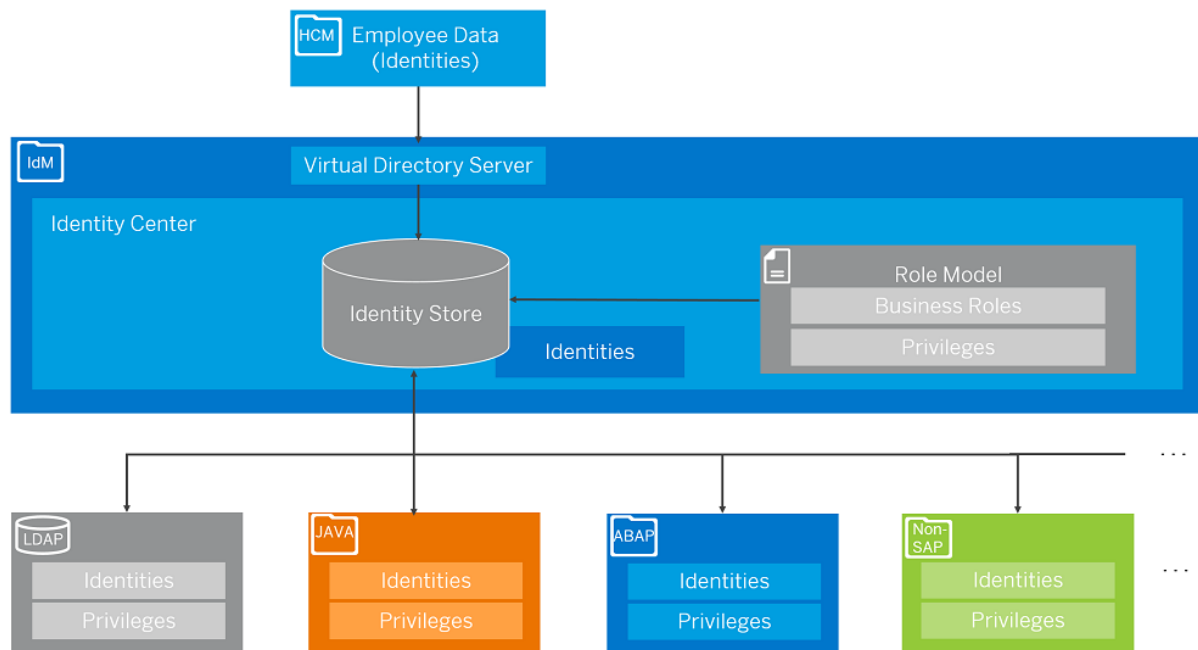
Local changes to other attributes with a different source system, for example, e-mail addresses where the source system is a mail server, or changes to role or role assignments can be provisioned back to the corresponding source system.

- Assign the business roles to identities in the Identity Center according to the role model. The meaning of the business roles are then mapped to the corresponding privileges and the corresponding user role assignments are provisioned to the target systems.
- This use case shows how to set up identity management for the portal and AS Java system with one back-end AS ABAP system. You can use the same concepts to set up provisioning to further AS Java and AS ABAP systems. You can also provision to non-SAP systems.

2.4.5 Identity Lifecycle Management

In the last sections, we showed how SAP NetWeaver Identity Management works in a couple of SAP system landscapes. You can also expand these concepts to include further SAP systems, or non-SAP systems to further integrate identity management in your complete landscape.

See the figure below.



Use the same concepts as described for the SAP systems (SAP HCM, AS Java, AS ABAP) and the LDAP directory to add additional systems to your SAP NetWeaver Identity Management system. In this way, you can move to a holistic, heterogeneous identity management approach.

2.4.6 Enhanced SAP Business Suite Integration

Description

In addition to SAP HCM, you can integrate many applications from the SAP Business Suite into the SAP Identity Management landscape. In this case, application-specific processing such as the creation of a business partner is performed in addition to the provisioning of standard AS Java or AS ABAP identities (SU01 users) and their corresponding assignments. The corresponding connector is provided with the SAP provisioning framework.

Note

For many of the SAP Business Suite systems, for example, SAP CRM or SAP SRM, a central person is created and used to link an identity to his or her business partners. When an identity is created and provisioned with SAP Identity Management, this central person and the corresponding business partner is also created in the SAP Business Suite system.

Another enhancement available in this scenario is that certain communication data for the employee can be provisioned back to the SAP HCM system. This is not possible in the standard SAP HCM scenario. The table

below shows the applications that are supported by the AS ABAP for SAP Business Suite connector, additional application-specific release prerequisites, if applicable, and the feature provided for the application.

SAP Business Suite Systems and Features Supported with Enhanced Business Suite Integration

SAP Business Suite Application	Features	Prerequisites
SAP Human Capital Management	Sending of employee-related data from SAP HCM to SAP Identity Management. Transfer of identity data, including communication data, from SAP Identity Management to SAP HCM.	SAP HCM application component Personnel Administration as of SAP Enhancement Package 4 for SAP ERP 6.0
SAP ERP Financials (Auditing)	A user with the role SAP_PLM_AUDITOR will also receive authorizations for the transactions Audit Management and Audit Monitor, as soon as the user and authorization distribution has been completed.	CA-AUD (auditing) of SAP ERP cross-application components as of SAP Enhancement Package 4 for SAP ERP 6.0
SAP ERP Financials (Accounting)	A new SAP Financials user automatically receives access to all of the functions for the corresponding company code that apply to his or her responsibility.	A new SAP Financials user automatically receives access to all of the functions for the corresponding company code that apply to his or her responsibility.
SAP Transportation Management (SAP TM)	The combination of a user account, a business partner, and a central person is created automatically.	SAP HCM application component Personnel Administration as of SAP Enhancement Package 4 for SAP ERP 6.0 (optional) SAP TM 7.0 or higher
SAP Extended Warehouse Management (EWM)	The combination of a user account, a business partner, and a central person is created automatically.	SAP HCM application component Personnel Administration as of SAP Enhancement Package 4 for SAP ERP 6.0 SAP EWM 7.0 or higher with labor management activated
SAP Supply Network Collaboration	Triggers automatic generation of users and business partners for SAP SNC.	SAP HCM application component Personnel Administration as of SAP Enhancement Package 4 for SAP ERP 6.0 SAP SNC 7.0 or higher
SAP Service Parts Planning (SPP)	Triggers automatic generation of users and business partners for SAP SPP.	SAP HCM application component Personnel Administration as of SAP Enhancement Package 4 for SAP ERP 6.0 (for the creation of users and business partners for new employees)

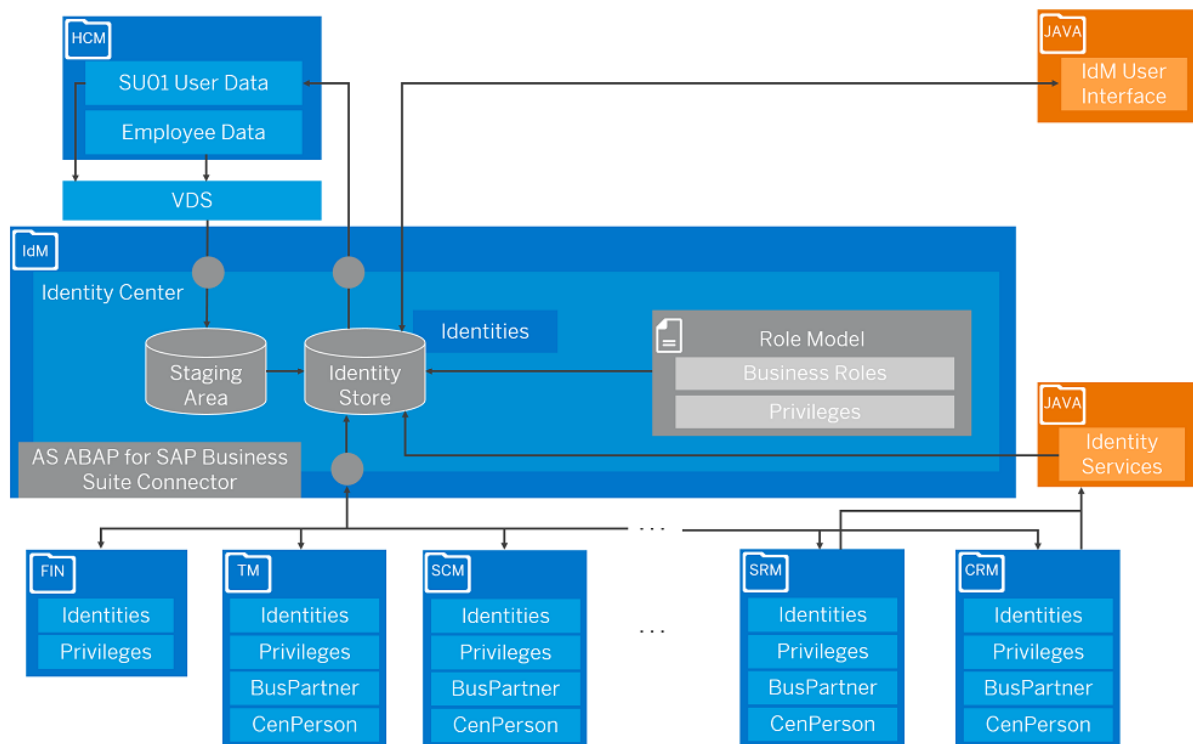
SAP Business Suite Application	Features	Prerequisites
SAP Product Lifecycle Management	Users are created in PLM, based on employee data from SAP HCM.	SAP HCM application component Personnel Administration as of SAP Enhancement Package 4 for SAP ERP 6.0 The PLM Web User Interface (PLM Web UI) is activated.
SAP Portfolio and Project Management	The combination of a user account, a business partner, and a central person is created automatically.	SAP HCM application component Personnel Administration as of SAP Enhancement Package 4 for SAP ERP 6.0
SAP Customer Relationship Management (SAP CRM)	SAP Customer Relationship Management (SAP CRM)	SAP CRM 7.0
SAP Supplier Relationship Management (SAP SRM)	The combination of a user account, a business partner, and a central person is created automatically.	SAP ERP HCM as of SAP Enhancement Package 4 for SAP ERP 6.0 SAP SRM 7.0

Technical System Landscape

The system landscape to use for this scenario is similar to the landscape for the other scenarios that involve SAP systems. Typically, the SAP HCM system is set up as the starting point for maintaining identity data, which is then provisioned to the target systems. The difference in this scenario is that the AS ABAP for SAP Business Suite connector is used to connect to the corresponding SAP Business Suite systems instead of the AS ABAP connector. This allows for the additional application-specific processing of the identity information.

In addition, certain SAP Business Suite applications (for example, by SAP CRM or SAP SRM) send identity-related information to SAP Identity Management using identity services, which run on an AS Java.

See the figure below.



Software Units

The following components are used in this scenario:

- Identity Center
- Virtual Directory Server (assuming the SAP HCM is included in the system landscape)
- Identity Management User Interface

The following connectors are used in this scenario:

- SPML connector (for AS Java target systems, or non-SAP systems that use SPML)
- AS ABAP for SAP Business Suite connector (for SAP Business Suite target systems)
- LDAP connector (for directory servers)
- Additional connectors (as appropriate for the target systems)

The following frameworks are used in this scenario:

- SAP provisioning framework
- SAP HCM staging area identity store
- SPML IDS identity store (for SAP CRM and SAP SRM applications)

Implementation Sequence

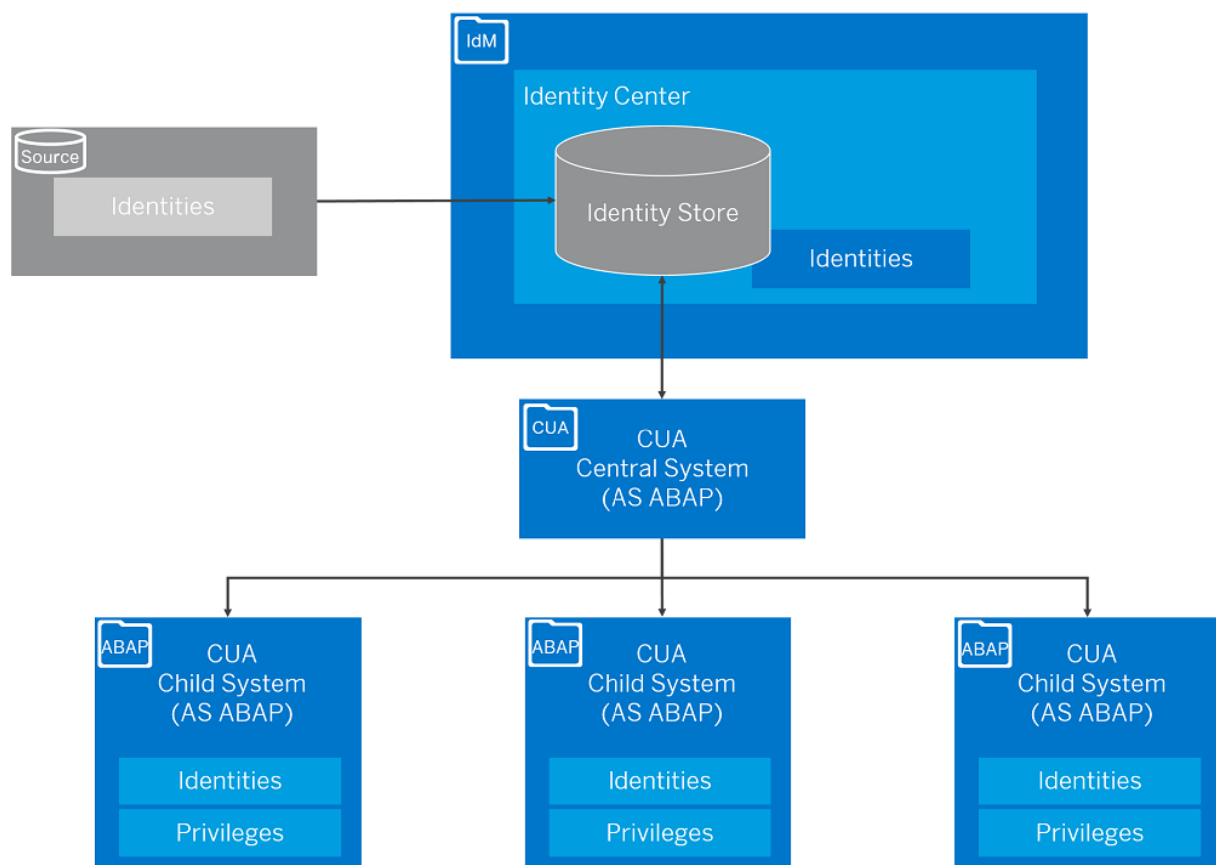
For an overview of the implementation sequence, see the Overall Implementation Sequence.

The following documents provide more information about enhanced SAP Business Suite Integration.

2.4.7 Integrating a Central User Administration System

As of version 7.0 SP02, SAP Identity Management supports the integration of a central user administration (CUA) system.

In this case, connect the CUA central system to the Identity Center as a target system in the same way as any other ABAP-based SAP system. The Identity Center provisions the identity data to the CUA central system, which in turn provisions the data to its child systems. This model works regardless of what system is used as the leading system for the identity management landscape. See the figure below.



This allows for a smooth installation of SAP Identity Management into an existing CUA landscape without any modifications, and at the same time, provides support for additional AS Java or other third-party systems.

You can also continue with migration steps to remove the child systems from the CUA landscape and connect them directly to SAP NetWeaver Identity Management.

Note

The CUA is not supported with the SAP Business Suite integration use case.

Note

Using both integration scenarios - SAP Identity Management with CUA and SAP Identity Management with SAP Governance, Risk and Compliance (GRC) is not supported. You can use only one of them in your system landscape.

Related Information

[Restrictions when Connecting a CUA System \[page 1334\]](#)

2.5 Concepts

2.5.1 Access Control

Per default, access to information in the identity store from the Identity Management User Interface is limited based on access control on tasks. If different groups of users (for instance several companies) share the same identity store, you may want to prevent the users of the different groups/companies to see each other's data. You can use the fields in the "Access limitations" group box to specify these access restrictions. The access limitations are a global setting that restricts which entries will be returned when a user searches for entries in the "Manage" tab and when adding references.

The mechanism relies on two attributes, the "Search attribute" and the "User attribute". The attribute defined as the search attribute must be defined on each entry of this entry type. In the simplest form the same attribute is defined as the user attribute. To define a user's access, the user's user attribute is compared to the entries' search attribute, and only those entries that have the same value as the user's user attribute are displayed.

Note

When using entry reference attributes, you can expect the following behavior:

- Access limitations apply for all entry reference attributes starting with MXREF_.
- Access limitations do not apply for other entry reference attributes, such as MX_MANAGER. For example, when you search for a user in SAP Identity Management User Interface, you can open the display form of the user's manager, regardless of the access limitations configured on entry type level (MX_PERSON). Displaying the manager's details will only be denied, if [Use access limitations for display task](#) option is selected on the entry type. In this case, you get Access Denied.

Caution

If the value of the attribute that is defined as a search attribute and a user attribute contains asterisk (*), this significantly slows down the search process. We recommend that you avoid using asterisk (*). Instead,

define the attribute as a multivalue attribute and add the corresponding values. In this case, the user's user attribute is compared to the entries' search attribute, and only those entries that have the same value as the user's user attribute are displayed.

For example:

- **DE** (Can see all German employees)
- **ES:Barcelona** (Can see all Spanish employees in Barcelona)

Note

Using access limitations will impact performance. For example when searching for roles and displaying role memberships, the access limitations must be checked for every role before displaying it.

Related Information

[Identity Store \[page 74\]](#)

[General \(Entry Type\) \[page 981\]](#)

2.5.2 Approval Processing

Note

The description of approval processing is based on the 7.2 approval mechanism introduced with SAP NetWeaver Identity Management 7.2 SP04.

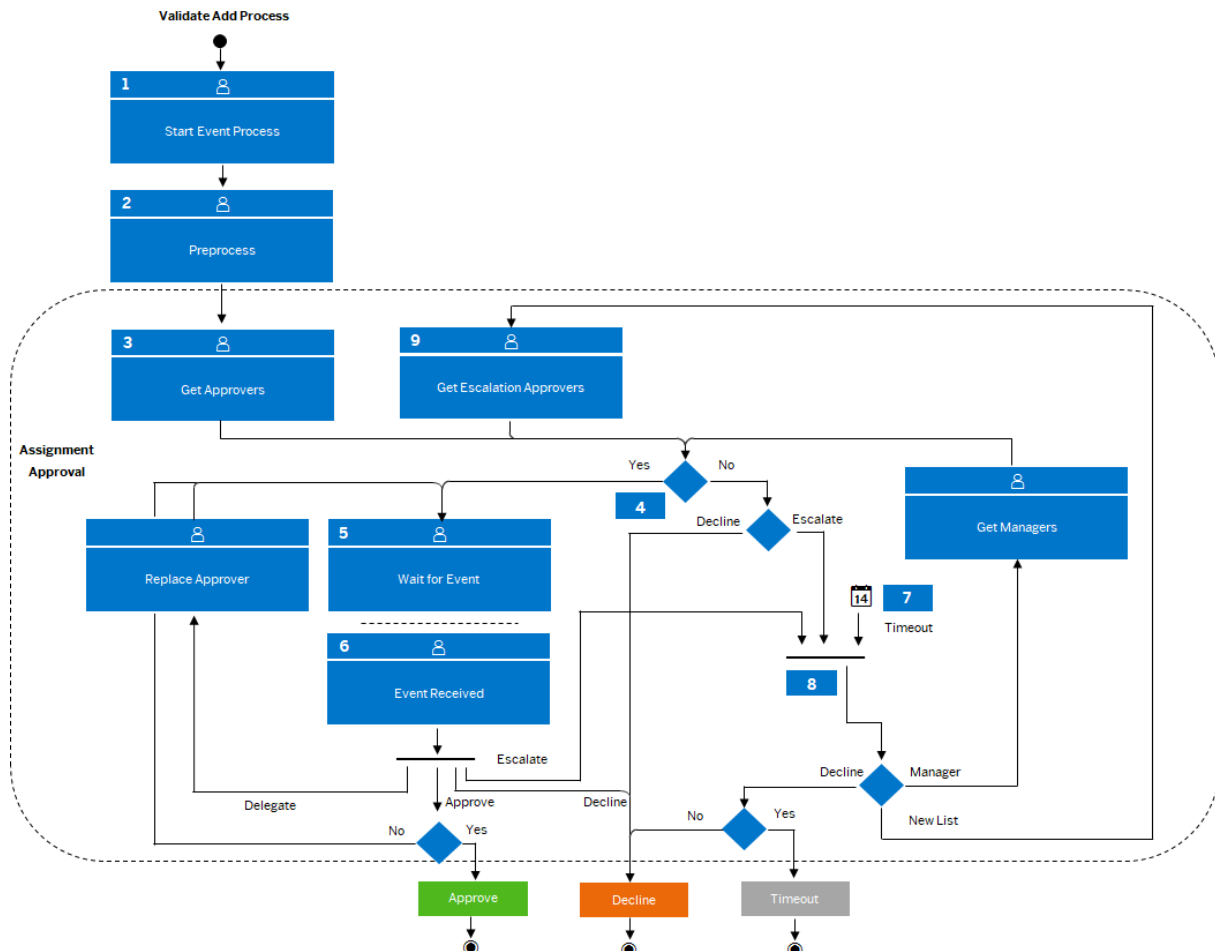
The processing of the approval request depends on the approval type:

- Assignment approval
Any approval which is started as a result of a member event on a role or privilege, for instance a Validate add process.
- Basic approval
Any other approval than for assignments, for instance approving a change done in the Identity Management User Interface, by adding the approval task inside a process.

The processing is defined in the Identity Management Developer Studio on the [Approval](#) tab of the approval task.

Processing an Assignment Approval Request

The following diagram shows how an assignment approval request is processed:



- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)

- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)

The example in this diagram is an approval request of an assignment (Validate add process).

1. When the event occurs (Validate add, Validate remove or Validate modify validity), the process referenced by the event is started. This can for instance be a process where the approval task is one of the sub-tasks.
2. Any pre-processing is done by tasks above the approval task within the process. If the pending value object is the source of the approval, you can for instance add additional approvers on the pending value object.
3. The approvers are identified. Based on the configuration on the approval task, they can be defined either on the role/privilege, the context, the generated pending value object or the approval task itself. The list of approvers is only calculated at this point. Although the conditions may change during the approval process, no approvers will be added or removed from the original list.

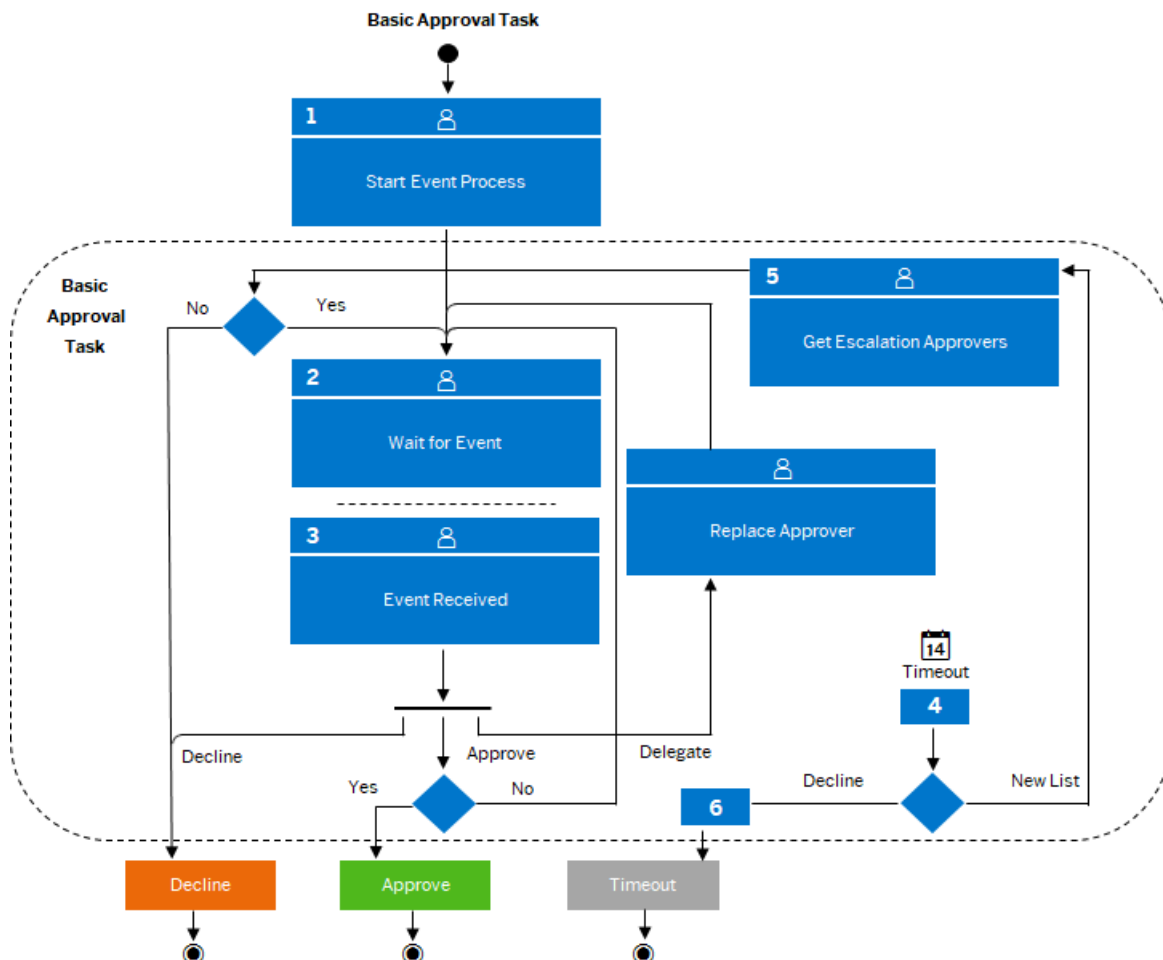
The following users are defined as not legal approvers, and will be removed from the list:

- The user getting the assignment
 - The user requesting the assignment
 - Any user that already has been involved in the approval process
 - A user that is not allowed to see the role or privilege due to visibility constraints
4. If no approvers are found or the number of identified approvers is less than the number of required approvals, the specified action for "Insufficient approvals" is used. This can either be *Declined* or *Escalate*. In case of *Escalate*, the action defined as *Escalation rule* will be used.
 5. When the approvers are identified, the approval is displayed in the *To Do* list of each approver, waiting for an approve or decline.
 6. The event is received:
 - *Approve*: When an approver approves the request, it is checked whether the required number of approvals is reached. If so, the assignment is done, and any tasks configured on the *Approve* node are executed and the approval task completes. It is not required to have any tasks under this node. The assignment is processed when it is approved by the required number of approvers. If not, it will continue to wait for approvals.
 - *Decline*: If one of the approvers or a manager/administrator declines the approval, any tasks below the *Decline* node are executed and the approval task completes.
 - *Delegate*: If one of the approvers delegates (forwards), the approval is removed from the original approvers *To Do* list and added to the list of the new approver and the approval continues to wait for an event.
 - *Escalate*: If a manager or an administrator manually escalates the approval, the approval is escalated according to the defined escalation rule.
 7. The task waits for the approvals for the given timeout period. If the timeout is reached before the required number of approvers has approved the request, the *Escalation rule* is checked.
 8. If the approval is escalated or times out, it is handled according to the escalation rule:
 - If the escalation rule is set to *Decline*, it is checked what is the reason for the escalation.
 - In case of a timeout or if the maximum number of escalations is reached any tasks below the *Timeout* node are executed and the approval task completes.
 - If the approval request is escalated manually or if it is escalated because there are insufficient approvers, the tasks below the *Decline* node are executed and the approval task completes.

- If the escalation rule is set to [Escalate to manager](#), the approval is forwarded to the managers of the remaining approvers. If any of these already have been approvers, they are removed from the list.
 - If the escalation rule is set to [Escalate to new list](#), the escalation approvers are found on the same object as the original approvers were defined.
9. The approval is then processed again, and the timeout is defined by the escalation timeout on the task. The number of escalations is limited by [Max no of escalations](#) defined on the task. When the approval is escalated more than this number of times, it will be declined.
- If, for any reason, the approval is not processed within 90 days from the initial request, it will be declined. To keep an approval active for a period longer than 90 days, you need to use the `MX_MAX_EXPIRY_TIMEOUT` global variable. For more information, see [Timeout](#) section in [Approval \(Approval Task\)](#) [page 1119]
- If expiration timeout of the approval task is set, it matches the Valid To date of the role/privilege assignment. If, for any reason, the approval is not processed from the date of the initial request to the Valid To date of the role/privilege assignment, the approval will be declined. In case the approval timeout is valid after the expiration timeout is reached, the approval task takes the value of the expiration timeout.

Processing a Basic Approval Request

The following diagram shows how a basic approval request is processed:



- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)

- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)
- [Approval Processing \[page 34\]](#)

1. A basic approval can be initiated in one of two ways:

- By having an approval task below a web-enabled task, approving the change done in the Identity Management User Interface.
- As an event process on an entry type or attribute.
Unlike the assignment approval, the change is already performed when the approval task is started, so you have to use the nodes of the approval task to decide which actions to take.
For instance, an approval task is used to validate a modified attribute in the Identity Management User Interface. The modified attribute value is written to an intermediate attribute when the user chooses [Submit](#). The approval task is started, and when the change is approved, the approval task copies the attribute value back to the actual attribute.

2. After a modification that requires an approval has been done, the approval task is started.

Note

For basic approvals, the approvers are not calculated when the approval is initiated. As it is possible to define complex SQL queries to identify approvers, this cannot be done at this time. This calculation is done when a user activates the [To Do](#) tab in the Identity Management User Interface, and any approvals where this user is identified as an approver, are displayed in his/her [To Do](#) list.

Alternatively, approvers can be calculated by calling the stored procedure `mvp_check_approvals` with input parameter the approver's mskey. It calculates the approvals for the current input mskey. For more information on how to execute this stored procedure using SAP Identity Management Developer Studio, see SAP Note [2974026](#) .

3. The event is received:

- [Approve](#): When an approver approves the request, it is checked whether the required number of approvals is reached. If so, any tasks configured on the [Approve](#) node are executed and the approval task completes. If not, it will continue to wait for approvals.
- [Decline](#): If one of the approvers declines the approval, any tasks below the [Decline](#) node are executed and the approval task completes.
- [Delegate](#): If one of the approvers delegates (forwards) the approval, the approval is removed from the original approvers [To Do](#) list and added to the list of the new approver and the approval continues to wait for an event.

4. The task waits for the approvals for the given timeout period. If the timeout is reached before the required number of approvers has approved the request, the escalation rule is checked.

If the escalation rule is set to [Decline](#), or the maximum number of escalations is reached, any tasks below the [Timeout](#) node are executed and the approval task completes.

If the escalation rule is set to [Escalate to new list](#), the escalation approvers are found on the same object as the original approvers were defined.

5. The approval is then processed again, and the timeout is defined by the escalation timeout on the task.

6. The number of escalations is limited by [Max no of escalations](#) defined on the task. When the approval is escalated more than this number of times, it will be declined.

7. If, for any reason, the approval is not processed within 90 days from the initial request, it will be declined. To keep an approval active for a period longer than 90 days, you need to use the

MX_MAX_EXPIRY_TIMEOUT global variable. For more information, see *Timeout* section in [Approval \(Approval Task\) \[page 1119\]](#)

Automatic delegation of approvals

You can specify that all approvals should be forwarded automatically to another user. This can be used if a user is absent for a period of time, or in any other situation where all approvals should automatically be forwarded to someone else.

Automatic delegation is done in the following cases:

- For any new approvals
- If an existing approval is escalated
- If someone performs a manual delegation to a person with automatic delegation enabled
- In any other circumstance where this user is added as approver

The approver can configure this from the [To Do](#) tab, or it can be done by updating the attributes MX_AUTODELEGATE_MSKEY and MX_AUTODELEGATE_MESSAGE of a user from a task or job.

If an automatic delegation fails, the approver is removed from the approval, and if there are not enough approvers left to handle the approval, it is declined.

Some reasons why an automatic delegation fails are:

- The user getting the assignment
- The user requesting the assignment
- Any user that already has been involved in the approval process
- A user that is not allowed to see the role or privilege due to visibility constraints
- There is a loop of automatic delegations

The processing of automatic delegations is done asynchronously, which means that if for example an approver performs a manual delegation to another user, the approver will get an immediate response that the delegation is performed. Automatic delegation is processed afterwards, which means that if the approval later is automatically delegated, it may result in the approval being declined.

Related Information

[Approval Task \[page 1117\]](#)

[Configuring the Approval \[page 257\]](#)

2.5.3 Attestation

You use attestation to periodically confirm users' access rights to critical resources.

Normally, users' access rights to critical resources are controlled by assigning specific roles and there is a risk if these roles are assigned to the wrong people. Assignment to these roles should periodically be verified by someone who is responsible for the resource or assignment.

The verification is done by executing an attestation process for each role or privilege. Note that for one given assignment, there is one (and only one) attester. The attester will only see the assignments for which he or she is responsible. The attester has the option to confirm or reject the assignments, as well as delegating the responsibility to another person.

All attestation operations are logged, so that they can later be audited.

The following attributes for a role or privilege are used to configure attestation of that role:

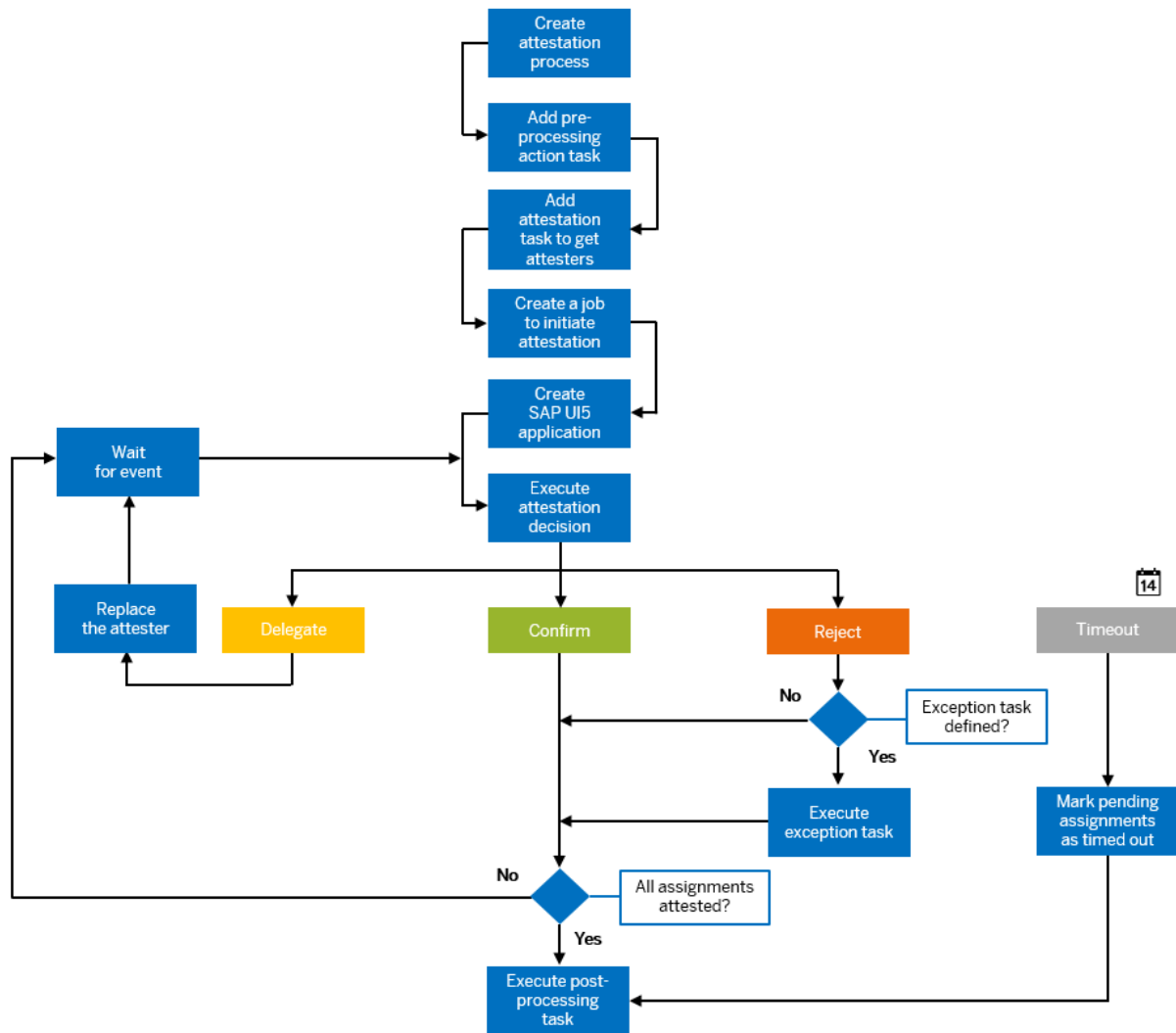
Attribute	Description
MX_ATTEST_TASK	Reference to the event process containing the attestation process to be executed.
MX_ATTEST_NEXTDATE	Date for the next attestation.
MX_ATTESTER	If the attestation process is configured to get the attester from the role or the privilege, the attester are added here.

For more information, see [Attestation Attributes](#).

These values can be added either by a job or from a task containing these attributes.

You can use calls from the REST Interface Version 2 to create an SAP UI5 application to process the attestation requests. For more information, see [SAP Identity Management REST Interface Version 2 \[page 828\]](#).

Attestation Workflow



Create Attestation Process

1. In SAP Identity Management Developer Studio, create an attestation process. See [Creating a Process \[page 220\]](#).
2. Make the process public.
3. Set the *Process Type* to *Regular*.
4. (Optional) Rename the process. For example: **AttestationProcess**.

Add Pre-Processing Action Task

The pre-processing task is optional. You can use it, if you want to get attesters from the *User Defined* option on the attestation task that you need to create in the next step. If *Get Attesters from* field on the attestation

task is set to *User Defined*, it is possible to define different attester for each assignment. This is done with a script in the pre-processing task that creates context variables containing the necessary information. For more information about context variables, see [Context Variables \[page 62\]](#).

To add a pre-processing action task, proceed as follows:

1. Open the **AttestationProcess** in the process flow diagram and add an action task to the process. See [Managing Process Contents \[page 225\]](#).
2. (Optional) Rename the action task, for example: **Pre-ProcessingTask**.
3. Open the **Pre-ProcessingTask** in the job view and add a *To Generic* pass.
4. Add a script calling the `uSetContextVar` internal function. You must create one context variable for each assignment. The name of the context variable is: `#MXATTESTER_<Linkid>`.

Context Variable	Value
#MXATTESTER_<Linkid>	The MSKEY of the attester.

The LinkID is the unique identifier for this assignment, and can be found in one of the views for reference attributes (see [Views for Audit of Reference Attributes \[page 139\]](#)). Use the MSKEY of the role to find the corresponding assignments ("where mcThisMSKEY = %MSKEY%"). This will result in a list of assignments for this role. Find the LinkID for each assignment in the column `mcLinkID`. How you find the MSKEY of the attester depends on who you want to be the attester. It could be determined based on a property of the assignment, or the user (found with `mcOtherMSKEY` in the view).

The following code is used to set one context variable:

```
OutString = uSetContextVar("#MXATTESTER_67", "43");
```

where 67 is the assignment's LinkID and 43 is the MSKEY of the attester.

If one assignment is to be skipped (and you do not want anything logged), set the MSKEY to **-1**. This can for example be the case if the assignment was done recently. If an assignment is missing, this will be logged in the link audit table. When the attestation is initiated, the attesters are found by reading this context variables.

5. Make sure you have selected *Use Context Variables* checkbox on the *General* tab of the **AttestationProcess**.

Add Attestation Task to Get Attesters

The configuration on the attestation task determines how to find the attesters.

To add an attestation task, proceed as follows:

1. Open the **AttestationProcess** in the process flow diagram and add an attestation task to the process just below the action task **Pre-ProcessingTask**. See [Managing Process Contents \[page 225\]](#).
2. (Optional) Rename the task, for example: **AttestationTask**.
3. Configure the **AttestationTask**. See [Configuring the Attestation \[page 267\]](#) and [Attestation \(Attestation Task\) \[page 1127\]](#).

Initiate Attestation

Attestation is initiated by a scheduled procedure or a job. The schedule procedure will check for the specified next date for the defined attestations, and start the event process referenced from the role(s). Optionally, attestation can be initiated by a job, with a script calling `uProvision` internal function. To initiate attestation by a job, proceed as follows:

1. In SAP Identity Management Developer Studio, create a job. See [Creating a Job \[page 289\]](#).
2. (Optional) Rename the job, for example: **TriggerAttestation**.
3. Add a *To Generic* pass.
4. Add a script calling the `uProvision` internal function. This function initiates a provisioning process for a given entry in the identity store.

The following code is used:

```
uProvision(19022, 987, 0, 0, "", 1);
```

where **19022** is the MSKEY of the role to attest; **987** is the ID of the public process containing the attestation task to be initiated; **0** is the reference audit, if available; **0** is the repository ID, if available; **""** is the general user ID or message that will be inserted in the field USERID in the audit log; **1** is the delay in seconds until the task should be initiated.

The example above states the following: Run task 987 on MSKEY 19022 and wait for 1 second before the task is initiated.

5. Add the script to the *To Generic* pass of the **TriggerAttestation** job. (In the *Next Data Entry* field on the *Destination* tab).
6. Schedule the job to run regularly.
After attestation is initiated, an attestation entry (one record) is created in the `mxi_attest_entry` table for all users assigned to the role or privilege. This is valid both for direct and indirect assignments.

Create SAP UI5 Application

Use the attestation URIs Identity Management REST Interface Version 2 to create an SAP UI5 application to process the attestation requests.

All attestation entries can be listed via a REST call. The list includes the information about the attested entry that is needed for an attester to decide whether to confirm, reject, or delegate the entry. For more information, see [Attestation Access URIs \[page 937\]](#).

Execute Attestation Decision

The attesters go through their list of attestation requests and execute attestation decision. Possible outcomes are:

- **Confirm:** The attester confirms the attestation request. Then, the attestation entry is deleted from `mxi_attest_entry` table. A post-processing task is triggered. It is responsible for doing any post-processing after all attestation requests for the role or privilege are completed. For more information, see below.

- **Delegate:** If one of the attesters delegates (forwards) the attestation request, the request is removed from the original attester's *To Do* list and added to the *To Do* list of the second attester. The attestation request continues to wait for an event, that is to be confirmed or rejected.

- **Reject:** The attester rejects the attestation request. It is checked if there is an exception task referenced from the attestation task.

If defined, the exception task is used to process rejected attestation requests. For example, the task can be used to de-assign any rejected assignments. It is called for each attestation request, not waiting for all attestation requests for the role or privilege.

To create an exception task, proceed as follows:

1. In SAP Identity Management Developer Studio, create a process. See [Creating a Process \[page 220\]](#).
2. Make the process public.
3. Set the *Process Type* to *Attestation Exception Process*.
4. (Optional) Rename the process, for example: **RejectionProcess**.
5. Reference the exception task (the **RejectProcess**) from the attestation task. That is, select the **RejectProcess** in the *Exception Task* field on the *Attestation* tab of the **AttestationTask**.
6. Depending on how you want to handle the rejection of a role or privilege assignment, you can add different action tasks to the exception task (the **RejectionProcess**).

ⓘ Note

By default, the attestation logic in SAP Identity Management creates a number of context variables. You can use them to implement your custom logic for handling rejected attestation requests.

7. Make sure to select *Use Context Variables* on the *General* tab of the exception task (the **RejectionProcess**) to specify that the context variables are created.
8. The exception task (the **RejectionProcess**) receives the following context variables:

Context Variable	Value
#ATTEST_ROLEMSKEY	MSKEY of the role or the privilege being attested.
#ATTEST_LINKID	The LinkID of the assignment being attested.
#ATTEST_COMMENT	Any comment (reason) given by the attester.
#ATTEST_ATTESTID	ID of the whole attestation process. All attestation requests in one attestation process share the same attestation ID.
#ATTEST_NEWSTATE	The state of the attestation request. The exception task only receives attestation requests that are rejected.
	2: Rejected

You can access these context variables with the `uGetContextVar` internal function. See the example script to access the context variables:

```
uWarning(uGetContextVar('#ATTEST_ROLEMSKEY'), '');
uWarning(uGetContextVar('#ATTEST_LINKID'), '');
uWarning(uGetContextVar('#ATTEST_COMMENT'), '');
```

```
uWarning(uGetContextVar('#ATTEST_ATTESTID'), '');
uWarning(uGetContextVar('#ATTEST_NEWSTATE'), '');
```

- To remove an assignment, you can use the LinkID provided in the context variable and create an action task with a [To Identity Store](#) pass. In this pass, you can identify the assignment with the LinkID and use any of the operators, for example, to remove the assignment.

Add Post-Processing Task

The post-processing task is responsible for doing any post-processing after all attestation requests for the role or the privilege are completed. This could for instance be to take an action based on the state of each attestation request, to reschedule the attestation (setting MX_ATTEST_NEXTDATE attribute for the role) or some other task.

The post-processing task is triggered once in the following cases:

- The attestation requests for the role or the privilege are completed.
- The attestation requests for the role or the privilege are not completed and the timeout is reached.

To add a post-processing task, proceed as follows:

- Open the **AttestationProcess** in the process flow diagram and add an action task to the process just below the attestation task **AttestationTask**. See [Managing Process Contents \[page 225\]](#).
- (Optional) Rename it, for example: **Post-ProcessingTask**.
- In the job view add a [To Generic](#) pass and a script calling the `uGetContextVar` internal function. The post-processing task receives the following context variable: `#MXATTESTSTATUS_<LinkId>` with information about the state of the attestation request for this assignment. You decide what action to take based on the state of the attestation request. See the example script:

```
uWarning(uGetContextVar('#MXATTESTSTATUS_36656'), '#MXATTESTSTATUS_36656');
uWarning(uGetContextVar('#MXATTESTSTATUS_36657'), '#MXATTESTSTATUS_36657');
```

The context variable returns the following possible states for the attestation request:

State	Description
1: Confirmed	
2: Rejected	
100: No attester found	This will be set if no attester can be found for the assignment. This can happen for the following reasons: - If Get Attesters from is set to User Defined and the LinkID is missing or wrong. - If Get Attesters from is set to Manager and the manager of a user cannot be found.

State	Description
101: Timeout	
102: Assignment removed	The role assignment has been removed from the user after the attestation process was initiated.

You can use the LinkID of the assignment to look up more information in the views for audit of reference attributes. See [Views for Audit of Reference Attributes \[page 139\]](#).

Handling Timeouts

Timeouts are handled by the post-processing task (mentioned above).

The post-processing task is triggered in the following cases:

- The timeout is reached before all requests are processed.
- The timeout is not reached, but all requests are processed (rejected or approved).

2.5.4 Attribute Ownership

It is possible to define attribute ownership using repositories. An owner repository can be defined for each attribute in the identity store. This attribute ownership can be used to define attribute precedence. The owner repository will have priority over the other repositories. Changes from repositories other than the attribute owner will not be written to the identity store. This can be illustrated with the following table:

Attribute Precedence

		Existing attribute owner		
		Owner	Other	None
Source repository	Owner	Update allowed	Update allowed	Update allowed
	Other	Update not allowed	Update allowed	Update allowed
	None	Update not allowed	Update allowed	Update allowed

When a single attribute is deleted, the table above is used to determine whether the attribute can be deleted or not. When the whole entry is deleted, all attributes are deleted, regardless of attribute ownership/precedence.

If the attribute cannot be updated due to precedence, you can configure how this situation should be handled for each attribute.

Each job writing to the identity store must include a reference to the repository that should be set as attribute owner. The Identity Management User Interface must be defined as a repository to be able to act as an attribute owner.

2.5.5 Attribute Validity

It is possible to define a validity period for attributes in the identity store. The validity is indicated using the ValidFrom and ValidTo properties where ValidFrom indicates when the attribute value will be added, and ValidTo indicates when the attribute value will be removed.

Note

It is possible to specify a "Time to live" for the attribute. If this is specified, the ValidTo will automatically be created when an attribute value is added, unless ValidTo is specified, in which case the specified value will be used.

You can define several validity ranges for the same attribute (i.e. January 1 - January 15 and February 1 - February 14). Overlapping ranges for the same attribute (i.e. January 1 - February 15 and February 1 - March 14) is only allowed for regular attributes.

These properties can be set for the attributes in a To Identity store pass using the attribute property syntax. For assignments the validity period can be specified in the Identity Management User Interface when assigning the role or privilege. For regular attributes, this can only be done from a job.

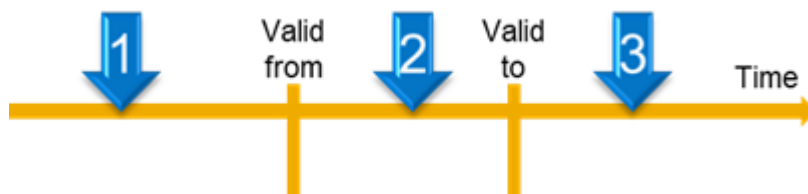
Note

Writing a value with validity does not change any of the existing validities. If you want to remove a validity range, this must be done explicitly.

2.5.5.1 Validity Operations

This description explains the behavior for the following operations depending on the situation when they are performed:

- Adding attribute with validity
- Modifying Valid from
- Modifying Valid to



The arrows 1, 2 and 3 illustrates alternative situations:

1. The specified Valid from and Valid to dates are in the future.
2. The specified Valid from date is in the past while Valid to is in the future.
3. The specified Valid from and Valid to are in the past.

This gives the following outcome:

Validity Operations

Operation	Situation when Operation Takes Place		
	1	2	3
Add attribute with validity	The attribute is created with Valid from and Valid to in the future. It will become active when Valid from occurs.	The attribute is created and is immediately active, as the Valid from date already has occurred.	The attribute is not created as the Valid to date has already been passed.
Modify Valid from	If the new Valid from is in the past, the attribute will be active. Otherwise, the Valid from date is updated.	If the new Valid from is in the past, the Valid from date is updated. Otherwise, the operation is not allowed, as the attribute is already active.	Not allowed, as the attribute is already active.
Modify Valid to	If the new Valid to date is later than the current Valid from, the Valid to date is updated. Otherwise, the operation is not allowed.	If the new Valid to date is later than the current time, the Valid to date is updated. For assignments, if the new Valid to date is earlier than the current time, the assignment is removed as the Valid to date is reached. Otherwise, the operation is not allowed.	Not allowed.

In any case, specifying a Valid to date that is before the Valid from will return an error.

2.5.5.2 Attribute Types

Validity is handled differently, depending on the attribute type.

The following attribute types can be defined:

- Regular attributes
Normal identity store attributes like title and address.
- Reference attributes
Attributes that are references to other entries in the identity store.
 - Assignments
A relation between a user and a role or privilege (with or without context).
 - Structural attributes
A relation between a role and another role or a role and a privilege, building the role hierarchy.

- Other reference attributes
References to for instance manager or secretary.

The attributes MXREF_<entry type> and MXMEMBER_<entry type> can be both assignment attributes and structural attributes in addition to indicating membership in a container (like MX_GROUP). This depends on the entry type of the entry that has the attribute:

Validity and Attribute Types

Attribute	Entry Type having the Attribute		
	MX_ROLE	MX_PRIVILEGE	Other entry type
MXREF_MX_ROLE	Structural attribute	Structural attribute	Assignment
MXREF_MX_PRIVILEGE	Assignment	Not allowed	Assignment
MXREF_<Other entry type>	Member	Member	Member
MXMEMBER_MX_ROLE	Structural attribute	Not allowed	Member
MXMEMBER_MX_PRIVILEGE	Structural attribute	Not allowed	Member
MXMEMBER_<Other entry type>	Assignment	Assignment	Member

In the following description structural attributes and other reference attributes are handled as regular attributes while assignments are described separately.

2.5.5.2.1 Regular Attributes

The following sections describe how validity is handled for regular attributes.

Future Values

When creating an attribute with ValidFrom set, a pending value object is created with the following attributes:

- MX_VALIDFROM
Date when the value is to be set.
- MX_VALIDTO
Date when the value is to be removed, or missing (if ValidTo is not specified).
- MX_ATTR_STATE
2 (Pending enabled, i.e. waiting for the ValidFrom date).
- MX_ATTRIBUTE_NAME
Name of attribute to set.
- MX_ATTRIBUTE_VALUE
Value of the attribute.

If only `ValidTo` is set, the attribute is set directly without any pending value object.

When ValidFrom is Reached

When the `ValidFrom` time is reached, the given value (`MX_ATTRIBUTE_VALUE`) is stored in the attribute (`MX_ATTRIBUTE_NAME`). If this is a single-value attribute, the existing value is overwritten. If it is a multivalue attribute, the new value is added. Any event tasks on attributes and entry types will be executed accordingly.

If `MX_VALIDTO` is present this will be set as the `ValidTo` of the attribute value.

Note

For regular attributes, the column in the views is called *ExpiryTime*.

When ValidTo is Reached

The housekeeping procedure `Check for expired attributes` verifies the validity of the specified `ValidTo` value of attributes. When the `ValidTo` value of an attribute is reached, the value will be removed. Any event tasks on attributes and entry types will be executed accordingly.

Modifying Validity

To change `ValidTo` of an existing attribute value, write the attribute with a new `ValidTo` property. To change validity of a future value, you must locate the pending value object, and update the `MX_VALIDFROM` and `MX_VALIDTO` attributes accordingly.

Removing Validity

To prevent a future value from being set, the corresponding pending value object must be deleted.

2.5.5.2.2 Assignments

The following sections describe how validity is handled for assignments.

Future Values

All assignments (also future assignments) are stored directly in the database, without a pending value object being created. Use the view `idmv_link_ext` to see any pending assignments. The columns `mcValidFrom` and `mcValidTo` contain the validity for the assignment.

Note

Overlapping date ranges are not allowed.

A pending value object will only be created if and when a task is to be executed for this assignment. See the topic [About member event handling](#) for details.

When ValidFrom is Reached

When `ValidFrom` is reached on an assignment, the member event tasks for this assignment are executed.

When ValidTo is Reached

When the `ValidTo` of an attribute value is reached, the value will be removed. Any member event tasks will be executed accordingly.

When `ValidTo` of an assignment is reached, the Remove member event process for this assignment is executed. In case, there is no connection to the back-end system, the assignment remains in Pending Delete status and the Remove member event process is triggered until the connection to the back-end is available. Then, the expired assignment is immediately removed.

Note

Regardless of its `ValidTo`, if a privilege assignment is removed from a user directly in the Identity Management UI when there is no connection to the back-end system, the deassignment process will not get into an endless loop in case the removal fails. It will be retried as many times as specified in the [No. of Retries](#) option, as described in [General \(Action Task\) \[page 1129\]](#)

Modifying Validity

Modifying validity of an assignment will cause the Validate Modify Validity task to be executed. Then the Modify Validity task is executed, and when executed successfully, the new validity is applied to the assignment. If the change is rejected, the existing validity is kept. The validity of an assignment can either be changed from the Identity Management User Interface or from a job.

You are able to change the validity of the attributes `MX_ASSIGNMENT`, `MXREF_MX_ROLE`, and `MXREF_MX_PRIVILEGE` via the button [Change Validity](#) in the Identity Management User Interface. The

utilization of the button enables you to create a request for a validity change of all selected direct assignments in the table simultaneously.

Note

The new button is active only for entry type MX_PERSON.

The button is enabled only when all selected assignments meet the following criteria:

- To be direct assignments with statuses **NEW** or **OK**.
- To be future assignments with status **PENDING**.

After clicking the [Change Validity](#) button, a window for validity change appears.

For simplicity, the valid from and valid to date fields will be automatically pre-filled in particular cases. For more information, see [3282726](#) .

Inherited Validity

The assignment of the parent is not made until the ValidFrom time, and at this time any child roles and privileges are assigned. If ValidTo is set on the assignment, this will be inherited to all child role and privilege assignments.

If the same child is inherited from several parents, the earliest ValidFrom and the latest ValidTo will be inherited. When removing one of the parents, the validity of the child will be recalculated, and will always be the newest/latest value.

Removing Validity

A future assignment must be deleted by specifying the following syntax in the To identity store pass:

```
{d}{VALIDFROM=aaa! !VALIDTO=bbb}PRIV
```

or by

```
{d}{LinkId=xxxxxx}PRIV
```

The assignment can also be deleted using the Identity Management User Interface.

2.5.6 Attribute DateTime Handling

This topic describes how each SAP Identity Management component handles attributes of type DateTime.

Note

Keep in mind that SAP Identity Management does not handle working with different time zones.

→ Recommendation

It is highly recommended to work only with UTC time. It is recommended that the configuration of the Database and Runtime servers includes one and the same time zone.

Database

On a database level, DateTime attributes are stored in a String format, following the UTC standard.

For example, **YYYY-MM-DD** or **YYYY-MM-DDTHH:MI:SS**.

Internally, even that the dates are stored in UTC format, they are interpreted as local Database Server timezone.

Runtime

While reading and processing the DateTime attributes from the database, the runtime considers the timezone of the local Runtime server.

While writing to the database, the runtime accepts the UTC format of the DateTime and values are written as a String to the database.

❖ Example

YYYY-MM-DD or **YYYY-MM-DDTHH:MI:SS**

SAP Identity Management User Interface

While reading the values from the database and presenting to the user interface, the WebDynpro component shows the DateTime that is stored on the database, no matter what timezone is configured on the NetWeaver server.

While writing to the database, the WebDynpro accepts the UTC format of the DateTime and values are written as a String to the database. The values passed should match the desired database or runtime server timezone.

❖ Example

YYYY-MM-DD or **YYYY-MM-DDTHH:MI:SS**

REST API V2

SAP Identity Management RestAPI v2 supports JSON timestamp format for read operations, and both JSON timestamp and String format for write operations.

While reading the values from DB, RestAPI returns a JSON timestamp object, the timestamp is in UTC timezone, matching the String value that is stored in the DB.

❖ Example

```
"SV_MX_VALIDFROM": "/Date(1609912800000)/"
```

While writing values to DB, RestAPI v2 accepts JSON timestamp object and also String representation of the DateTime. In case of JSON timestamp format usage, the value should be converted to UTC timezone prior to submit. In case of String format usage, the values passed should match the desired DB/runtime server timezone.

❖ Example

```
"SV_MX_VALIDFROM": "/Date(1609912800000)/"
```

```
"SV_MX_VALIDFROM": "YYYY-MM-DD" or "SV_MX_VALIDFROM": "YYYY-MM-DDTHH:MI:SS"
```

2.5.7 Audit Log

Whenever a new task is started, a new record in the audit table, with a new audit ID, is created for this task. This is called the root audit for the task. The ref audit column of the new record will be set to the same as the audit ID, to indicate that this is the root.

A running task may also start an event task, which may be done in many different ways. Some examples:

- By executing one of the result handling tasks (onOK, OnFail etc).
- By executing the initialize task (task group/action task).
- By changing attributes or entries which have event tasks (add, modify or delete).
- By assigning an "Add" or "Remove" member event task to a role or privilege.
- By assigning or removing a privilege, and the corresponding provisioning or deprovisioning task is started.
- By modifying an entry which has a privilege with a modify task.

In either case, an event task is started, which will get a new audit record and a new audit ID. In this case, the ref audit column is set to reference the parent task. In this way a hierarchy of audit records are created, as the tasks are executed.

A special case is calling uProvision function, where it is possible to specify which reference to set. This should normally be set to the parent audit ID, but if used with care, this can be set to link the task to any audit ID.

"Wait for event tasks"

You can specify *Wait for event tasks* on task execution and result handling. In this case, the ref audit field is used to determine whether tasks are still running. If one of the options is selected, it waits as long as there are tasks in the system where the ref audit is the same as the own audit ID.

The mxp_audit Table

The following table contains some selected columns of the mxp_audit table and how they are used:

The mxp_audit Table

Column	Description
AuditID	Unique identifier for each record in this table.
TaskId	Task which this audit record belongs to.
Mskey	MSKey which this audit record belongs to.
PostedDate	Date when this audit record was created.
StatusDate	Date when status was last updated.
ProvStatus	Status of the audit record. This can have one of the following values. In general, values ≥ 1000 means that the processing is completed. -1:Task on soft hold. Entry is locked by another task. 0:Task initiated OK 1:Task not enabled for provisioning 2:Task does not exist 3:Loop detected 4:Task cannot be called externally as it is private 5:Entry does not exist in Identity Store 10:Database error 20: Pending 21: Running 1000:Task OK 1001:Task failed 1100:OK 1101:Failed

Column	Description
RefAudit	This is a reference to the parent audit ID if this audit record was created as a result of the processing of another audit record.
MSG	Error message.
UserID	General information field. See below.

The UserID Field

The following is a formal description of the UserID field of the mxp_audit table:

```

UserId      ::= UserMsKey | OnEvent | AttrEvent | EntryEvent
UserMsKey   ::= Number -- MSKEY of user performing the operation
OnEvent     ::= "+" TaskId ":" TaskName
TaskId      ::= Number -- Id of task
TaskName    ::= String -- Name of task
AttrEvent   ::= "#" AttrId ":" Operation ";" CheckSum ";" OldValuesId
AttrId      ::= Number -- ID of attribute
CheckSum    ::= Number -- MD5 hash of entry
OldValuesId ::= Number -- Reference to mxi_old_values.old_ID
EntryEvent  ::= "*" EntryId ":" Operation
EntryId     ::= Number -- ID of entry
Operation   ::= "INSERT" | "MODIFY" | "DELETE"

```

2.5.8 Assignment Grouping

Overview

When performing an assignment which results in multiple privileges (for example, assigning a business role with many privileges), the member event processes (Validate add/Validate remove and Add member/Remove member) of each privilege will be executed individually, causing multiple requests to be sent to the target system. To prevent this, you can group assignments to reduce the number of requests sent to the target system.

There are two prerequisites for assignment grouping:

- The privileges belong to the same repository.
- The privileges are assigned within the same session (request).

Assignment grouping is configured as part of the repository properties. For more information about the grouping rule that can be defined on a repository, see [Grouping of Privileges \[page 744\]](#).

When assignment grouping is enabled:

- One pending value object is created for each assignment. Pending value objects that belong to the same group are linked together using the attribute `MX_PRIV_GROUPING_GUID`, which will have the same value (unique identifier) for all PVOs in the same group.
- Only one member event process will be started for the group, referencing one of the pending value objects in the group. This means that all the assignments within one group (having the same grouping GUID) will have one and the same AuditID.

📘 Note

Due to the specifics and the design of the Approval mechanism, we do not support the combination of grouping and approvals and we do not recommend using them together. However, if you decide to combine them, some faulty behavior may be observed. For more information, see [3196414](#) 🐞.

The grouping mechanism applies also to processes like reconciliation, provisioning, deprovisioning, and others.

📘 Note

In the SAP Identity Management User Interface, you can see the status of the assignments. If one or more assignments of an assignment group have failed, you can retry the assignments by clicking on the status link, and then choosing [Edit/Retry](#). Keep in mind that this action retries only the specific assignment that you have selected, and not the whole assignment group.

Functional and Performance Grouping

There are two types of assignment grouping depending on whether or not privilege assignments require compliance checks to the target systems (using SAP Identity Management - GRC integration scenario). Functional grouping applies only to privilege assignments that require compliance checks, while performance grouping applies to privilege assignments that do not require compliance checks.

📘 Note

Privileges that require compliance checks to the target systems have the `MX_AC_ROLEID` attribute defined on them. This is the GRC attribute that is used to differentiate the functional grouping from the performance grouping.

- **Functional grouping (GRC only)**

When you assign a number of privileges to a user (directly or through one or more business roles), all privileges that require compliance checks are grouped together for one request to the GRC system. The request is sent only if all member event processes, approvals or future validities that are defined on the privileges are completed.

If a privilege, that requires a compliance check, wait for 10 days to be approved, other privileges (assigned within the same session/request), that require a compliance check, will also wait for 10 days, until all privileges are finally grouped for one request to the GRC system.

- **Performance grouping**

When you assign a number of privileges to a user (directly or through one or more business roles), all privileges that do not require compliance checks are grouped depending on the grouping rule configured on the repository.

Unlike functional grouping, with performance grouping privileges that wait for an approval or have a future validity defined on them will not be grouped together with privileges that should be assigned immediately. With performance grouping, fewer requests are sent to the back-end system.

Note

With performance grouping, you cannot expect a fixed number of requests to be sent to the back-end system even though you have equal conditions. However, you can expect fewer requests to be sent to the back-end system.

Assignment Grouping in SAP Provisioning Framework

Note

Assignment grouping is only supported for AS Java, ABAP (ABAP BusinessSuite, Dual Stack) and GRC systems.

When using the assignment grouping in the above mentioned supported systems, have in mind the following specifics:

- **AS Java repositories**

If assignment grouping is defined for AS Java repositories, it is mandatory for the assignments to be separated by the privilege attribute MX_PRIVILEGE_TYPE. You should always select the [Separate by Privilege Attribute](#) option and set its value to be the MX_PRIVILEGE_TYPE attribute, regardless of what other options might have been selected. Otherwise, MX_ROLE and MX_GROUP objects might be sent with one request to the back-end system and this will cause the AS Java [AssignUserMembership](#) plugin to process only one of the two types of objects.

- **ABAP repositories and GRC**

Note

Assignments that require compliance checks before being provisioned to ABAP repositories are grouped together for one request to the GRC system but once approved, they are grouped and provisioned separately to the ABAP repositories.

Custom Assignment Grouping

The member event process must be configured to process the data in all the pending value objects in the group. Typically, you would do the following in the first script executed in the task:

- Obtain the MSKEY of the pending value object that the task is started for
Example:
`Use CurrentEntryID = uGetEntryID();`
- Obtain the value of this pending value object's attribute MX_PRIV_GROUPING_GUID
Example:
`GroupGuid = uIS_GetValue(PendingMSKEY , CurrentIDStore, "MX_PRIV_GROUPING_GUID");`

- Obtain the list of all other pending value objects with the same GroupGuid

Example:

```
var SQL = "SELECT mskey FROM idmv_value_ext_active WHERE
attrname='MX_PRIV_GROUPING_GUID' and searchvalue='"+GroupGuid+"'";
```

From these pending value objects, you normally need to retrieve all necessary information in order for your task to function properly. This data can be stored in context variables that can be read later:

Example:

```
OutString = uSetContextVar("CONTEXT_VAR_NAME", ContextVarValue );

aValue = uGetContextVar("CONTEXT_VAR_NAME ", "default value");
```

2.5.9 Context Based Assignments

To any role or privilege assignment, it is possible to add a reference to a given context that limits the validity of the assignment to that specific context. The only exception is the account privilege **PRIV:<Repository>: ONLY**, which cannot be assigned with context.

A context may be a region, a project or an organizational unit. The purpose is to reduce the number of roles. This can for example be used in the following scenarios:

- Project management
In this case the context is a project. A limited number of roles are defined, for example project manager, test manager, development manager, marketing manager etc. A new entry type is defined for the project. Each project is defined as entries of this entry type. An assignment will then consist of the combination of a role and a project.
- Store management
In this case the context is a store, and there will be a limited number of roles which are common to all the stores.

The specified context will follow the inheritance of the assignment. If a role is assigned with a given context, all inherited roles and privileges will also be assigned within the same context.

The roles or privileges are assigned using an assignment request task.

For more information, see the document [SAP Identity Management Tutorial: Context-Based Assignments](#) 

Note

One assignment can only have one context.

Defining the Context Types

Each type of context must be created as a separate entry type, for instance Region or Project. Each context is defined as an entry with this entry type. For instance if the context type is Project, each project is defined as an entry with this entry type.

Repositories, Privileges and Roles

The main purpose of implementing context based assignments is to reduce the number of roles (or privileges) required to map the authorizations or rights in the external (backend) system. For instance, if a project manager has different rights based on which projects she manages, the combination of role and context (project) will be mapped to different authorizations in the backend system.

The backend system is updated using member event tasks, typically defined as part of the repository definition. An Add member event task that, for instance, creates an account in the backend system receives references to the user, the role and the context, and based on that information, it is responsible for assigning the correct authorizations in the backend system.

The repository definition includes a number of legal contexts types, which will be used for all privileges referencing this repository definition. The context configuration can be overridden by defining the legal context types as part of the privilege properties.

The legal context types must also be defined for the roles that use context.

The legal context types are used for two purposes:

- The Identity Management User Interface will use this information when performing a direct assignment. A search for context will only be done in the defined entry types. When searching for a role or privilege, only the roles/privileges that are relevant for this context type are shown.
- During inheritance, only roles or privileges that supports the same context type as specified in the assignment are inherited.

Unless privilege grouping is defined for the repository, the event tasks on the role or privilege is executed for each context for an assignment, and each has a separate approval, execution and status.

Conditional Contexts

For privileges, it is also possible to define a list of conditional contexts. In this case, the privilege will only be assigned if one of the specified contexts also is part of the assignment. For instance, when assigning a role with a region as context, only those privileges matching the given region will be assigned.

Default Contexts

It is possible to define default context(s) for each user by adding the attribute MX_CTX_AUTO_VALUES to the person entries. This attribute contains the default contexts for each context type. For instance, this can be used

to define a default region for each user. This value is then used when assigning a role or privilege that uses region as context type. Only default contexts for the context type in question will be used.

The context configuration for a privilege or repository includes the policy for how this value is to be used.

The check is performed for both direct and inherited contexts.

Performing an Assignment with Context

To be able to make an assignment including a context, the role or privilege must have the entry type of the context object as a legal context type. This is done by setting the MX_CTX_TYPE on the entry to the entry type you want to use as context.

Each and every role and privilege which you want to assign with a context must have the MX_CTX_TYPE set. If you attempt an assignment with a context entry, where the entry type of the context entry is not defined in the MX_CTX_TYPE:

- If this is a direct assignment, an error is returned
- If this is an indirect assignment, this is silently ignored, and you will not get the inherited role/privilege

It is not recommended to remove the MX_CTX_TYPE from a role or privilege once assigned. There is no automatic reconciliation process which will detect this, as the context types are assumed to be unchanged. If the MX_CTX_TYPE is changed, do not make any modifications on the given user until the reconciliation process is finished.

2.5.10 Context Variables

The purpose of the context variables is to have variables which are transferred between tasks within the same process. A context variable will always belong to one context (audit ID). When the execution thread terminates, the context variables are automatically deleted.

This means that one task can add a context variable, and another task (within the same context) can read and/or modify the context variable.

You specify that context variables starting with "#Mx" will be used on the [General](#) tab of the process properties.

ⓘ Note

The maximum length of the context variable name is 255 and for the value it is 2000 characters.

ⓘ Note

Context variables are not set from the Identity Management User Interface if it is run on SAP NetWeaver AS for Java 7.0.

ⓘ Note

If you are writing context variables from the Identity Management User Interface, all single value mandatory attributes (including MSKEYVALUE and DISPLAYNAME) are always listed as modified attributes.

u-Functions

The following functions are available in the runtime engine for handling the context variables:

- [uSetContextVar \[page 1221\]](#)
- [uGetContextVar \[page 1161\]](#)
- [uIncContextVar \[page 1174\]](#)
- [uAppContextVar \[page 1139\]](#)
- [uMGetContextVar \[page 1199\]](#)

Pre-Defined Context Variables

A context variable starting with "#Mx" is reserved and cannot be modified. The only exception to this is #MxInherit_All.

By setting the option [Use Context Variables](#) on a process, a set of pre-defined context variables will be added.

Note

In case that you customize a parent process (which includes child processes in it) that links the [Modify](#) process of the [com.sap.idm.provisioning.engine](#) package, the [Use Context Variables](#) checkbox on the [General](#) tab of the process must be selected.

Context Variables

Context Variable	Value
#MxEventType	The type of event which started this task. This can be one of the following: 1: EntryAdd 2: EntryModify 3: EntryDelete 4: AttributeAdd 5: AttributeModify 6: AttributeDelete 7: OnOK 8: OnError 9: Provision task 10: Deprovision task 11: Modify privilege task 12: InitTask 13: OnChainOK 14: OnChainError 15: AddMember task 16: DelMember task 17: NoMaster task 18: ValidateAdd 19: ValidateDel 20: ModifyValidity 21: Validate Modify validity 22: Notify 23: RequestComplete 24: IdMUI submit
#MxEventAttrCount	Number indicating how many attributes where changed. This is used in the naming of the following attributes. If for example 3 attributes are modified, there will be #MxE-ventAttrName_1, #MxEventAttrName_2 and #MxEventAttrName_3, with the corresponding values. Multi-value attributes will be added as multiple variables in the same way.

Context Variable	Value
#MxEventAttrName_n	Attribute name which started the task.
	 Note This must be the name and not the ID of the attribute, to be able to handle configuration staging.
#MxEventAttrValue_n	The new value of the attribute. Empty if operation was delete.
#MxEventAttrOldValue_n	The previous value of the attribute. Empty if operation was add.
#MxEventAttrContext_n	The new context MSKEY of the attribute value. Only relevant for context based privilege and role assignments.
#MxEventAttrOldContext_n	The previous context MSKEY of the attribute value. Only relevant for context based privilege and role assignments.
#MxEventAttrtype_n	Used for modify entry task, indicating if the attribute was added, modified or deleted.
#MxParentAuditId	Audit ID of parent.
#MxRepositoryId	Repository identifier.
#MxParentTaskId	ID of task which started this task.
#MxMasterPrivilege	MSKEY of the master privilege. Set when #MxEventType is 17 (NoMaster task), to indicate which privilege is missing.
#MxOldValidFrom	Used in Modify validity task. Contains the old Valid from date.
#MxOldValidTo	Used in Modify validity task. Contains the old Valid to date.
#MxNewValidFrom	Used in Modify validity task. Contains the new Valid from date.
#MxNewValidTo	Used in Modify validity task. Contains the new Valid to date.
#MxLinkReference	Unique numeric reference to the link table.
#MxRequestID	RequestID. Used with the Request complete task. Set when #MxEventType is 23.

Context Variable	Value
#MxInherit_All	A flag variable used to denote whether context variables starting with #Mx will be copied to a process that is triggered by uProvision. If set to 1, all context variables starting with #Mx will be copied to the child process from the parent process together with user created context variables.

Note

When a process is started as an event of a parent process, #Mx context variables are copied to the child process from the parent process only if [Use Context Variables](#) option is set on both processes. When a process is started by the uProvision function, #Mx context variables are copied to the child process from the parent process only if #MxInherit_All is set to **1**.

Note

Any attributes which are larger than around 2K will not be copied to the context variables.

Note

If the [Use Context Variables](#) option is not set on the process, no #Mx context variables will be created. This is due to performance, to not create context variables unless they are used.

Example

You have a [Change Identity](#) form that is used to modify users' first, and last names as follows. When you submit this form, the following processes will start in parallel: a process that is defined on the [Result Handling](#) tab of the form as [Execute Process on Submit](#), and the corresponding event processes that are defined on the attributes when attribute values are modified.

These will have the following context variables if [Use Context Variables](#) option is set on all processes and the form:

Context Variables Examples

Context Variable	Value	Comment
#MxEventType	2	Indicates that this is a modify event.
#MXEventAttrCount	2	Two attributes were modified.
#MxEventAttrName_1	MX_LASTNAME	Attribute name of the first attribute.
#MxEventAttrType_1	Modify	The attribute was modified.
#MxEventAttrValue_1	Doe	The new value of the first attribute.
#MxEventAttrOldValue_1	Hero	The original value of the first attribute.

Context Variable	Value	Comment
#MxEventAttrName_2	MX_FIRSTNAME	Attribute name of the second attribute.
#MxEventAttrType_2	Modify	The attribute was modified.
#MxEventAttrValue_2	John	The new value of the second attribute.
#MxEventAttrOldValue_2	Jane	The original value of the second attribute.

2.5.11 Dispatchers

The dispatcher is installed as a separate (Java) program on all computers together with the runtime engine. It can be installed as a service on computers running Microsoft Windows or as a daemon on Unix.

The purpose of the dispatchers is to:

- Start the runtime engine when a job is to be processed.
- Perform the task execution logic, such as conditional tasks.
- Evaluate relations (assignments) and approvals.
- Execute the scheduled procedures used for housekeeping.

A dispatcher is continuously connected to the Identity Center database and checks for scheduled jobs. As soon as a job is ready to be run, the dispatcher will detect this and start the runtime engine that processes the job.

The dispatchers are also responsible for processing the tasks in the provisioning system. When a task is initiated through explicit or implicit provisioning, it is placed in a queue. Any dispatcher that connects to the Identity Center will process any task(s) that are in this queue.

A job configuration contains a reference to one or more dispatcher(s) responsible for executing the job when it is scheduled to run. A dispatcher can be configured to handle jobs for the Java runtime engine. It can also be configured to run provisioning jobs, regular jobs or both.

Each time the dispatcher is checking for scheduled jobs, it writes a timestamp in the database. This table can be monitored, to detect whether a dispatcher is no longer running and take the appropriate actions.

You can configure options that are common to all defined dispatchers.

At least one dispatcher must be configured to execute the scheduled procedures that are defined. The dispatcher log contains log entries for each execution of the procedures.

The rule is to have only one dispatcher in a system. The exception to this rule is:

- Set up an identically configured dispatcher on separate hardware for high-availability reason. This also requires that the database is set up for high-availability.

For more information about how many dispatchers should be configured for an SAP Identity Management system, see [2437631](#) .

For more information about how you configure the dispatchers for optimal performance in a production system, see *SAP Identity Management Identity Center Implementation Guide: Optimizing dispatcher performance* available on the SAP Community Network (SCN).

Related Information

[Maintaining Dispatchers \[page 385\]](#)

[Dispatcher Properties \[page 974\]](#)

2.5.12 Encryption

Hashing and encryption is used to protect information from being viewed and read. Typical examples are passwords and other sensitive information to be protected from unauthorized users and processes. For detailed information about how to implement encryption within *SAP Identity Management*, see the *SAP Identity Management Security Guide*.

Hashing

Hashing is used in the following cases:

- Authentication (from the Virtual Directory Server (MX_PASSWORD) and the password recovery (MX_AUTHQ_nnn))
- Internal authentication in the Virtual Directory Server
- Data hashed with the internal function uGenHash
- Any attribute can be configured to use the defined hashing algorithm. When an attribute is hashed, so are all historic values for the given attribute. However, changing the hashing algorithm on an existing attribute does not change any values in the identity store.

There are the following hashing algorithms:

- SSHA (Salted SHA1)
- SHA256
- SHA384
- SHA512
- PBKDF2_SHA1

What is the default hashing algorithm is defined as part of the options of an Identity Center.

Encryption

The following information can be encrypted:

- Connection strings, used to connect to the Identity Center database or other repositories. These are always encrypted.
- Passwords, which are stored in configurations. These are always encrypted.
- Job constants and package constants can be encrypted if desired.
- Any attribute can be configured to use the defined encryption algorithm. When an attribute is encrypted, so are all historic values for the given attribute. However, changing the encryption setting on an existing attribute does not change any values in the identity store.
- Any data encrypted with the internal function `uEncrypt`.

There are the following options for encryption:

- DES3/CBC
- AES128/CBC
- AES192/CBC
- AES256/CBC

What is the default hashing algorithm is defined as part of the options of an Identity Center.

What can be Encrypted?

The following information can be encrypted:

- Attributes in the identity store
These are encrypted according to the encryption algorithm specified on the [Storage](#) tab of the [Identity store attribute](#) dialog box.

Note

A job writing to the identity store must perform an explicit encryption, using the internal function `uEncrypt`.

- Connection strings/JDBC URLs
The connections strings/URLs used to access the Identity Center database are always encrypted using the default encryption algorithm.
Connection strings/URLs to databases that are accessed from the pass definitions can be encrypted if [Encrypt connection string](#) is selected.
Connection strings written to the files:
 - `dispatcher_service_<name>.prop`
 - `eventagentservice_<name>.prop`
 - `emsconfig.xml`are always encrypted using the default encryption algorithm.
- LDAP passwords
These are always encrypted using the default encryption algorithm.
- Package constants (Identity Center/Standalone jobs)

Package constants can be encrypted using the default encryption algorithm if *Encrypt value* is selected in the *Package constant* dialog box.

- Repository constants
When defining a repository, passwords are encrypted using the default encryption algorithm.
- Job constants
Job constants can be encrypted using the default encryption algorithm if *Encrypt value* is selected in the *Job constant* dialog box.
- Data
When creating a pass, the data can be hashed/encrypted using one of the internal functions uGenHash and uEncrypt (uDESEncrypt, uMD5, uSHA1).

Note

With the exception of attributes in the identity store and when using explicit functions, all encryption is done using the default algorithm specified in the *Options* dialog box.

Using Triple DES Encryption (DES3/ECB and DES3/CBC)

Note

The Java Cryptographic Extension Jurisdiction Policy files are a prerequisite for the SAP JVM must be downloaded separately as described in Note 1240081.

An external key file, keys.ini, is used for encryption. The default location of this file is:

<installation directory>\key\keys.ini

For details about how to maintain this file, see the *SAP Identity Management Security Guide*.

Decrypting Data

Decryption is done using the same algorithm as for encryption. If DES3/CBC is used, the module performing the decryption must have access to the encryption key.

With the exceptions below, a runtime engine will automatically decrypt any encrypted data (including configuration data).

- Hashed data cannot be decrypted, as these are one-way encryption algorithms.
- If the encryption key is unavailable.

Related Information

[SAP Identity Management Security Guide](#)

2.5.13 Entry Types

Entries in the identity store can be used to store information about different types of objects, such as persons, roles, privileges and locations. An entry type defines which attributes are allowed and which attributes are mandatory. The entry type also contains other information that will be common for all entries of this type.

When you create an identity store, a number of entry types are created automatically, such as MX_PERSON, MX_ROLE and MX_PRIVILEGE. In addition, you can select which additional entry types to import. You can also import entry types later or create your own entry types manually.

When creating an entry, the entry type should always be specified. This can be specified both for a task (when creating entries from the Identity Management User Interface) and in a To identity store pass. When creating an entry either with a job or from the Identity Management User Interface, the entries will be given this entry type.

When new entries are created, the list of mandatory and optional attributes is checked. If this does not conform to the definition, an error is returned, and the entry is not created.

An entry can have only one entry type. The attribute MX_ENTRYTYPE contains the entry type for the entry. Although possible, it is not recommended to change the entry type of an entry. When changing (or adding when missing) the entry type, it will be verified that all mandatory attributes are present, and that all attributes are defined as allowed for the entry type. If the verification fails, the update is not done.

Related Information

[Identity Store \[page 74\]](#)

2.5.14 Forms

Forms define the user interface which the end-users can use to modify or add data to the identity store. A form holds the following:

- Attribute definitions, meaning which attributes are shown and in which order
- Access control, meaning who can execute the form
- Other user interface configuration elements, such as display name, header, footer and so on
- Optional tasks, which are executed after the user selects *Submit*

In addition to the normal forms that can be accessed from the Identity Management User Interface, there are also some guided activities, which are intended to guide the user through more complex procedures. There are several types of guided activity forms available for different purposes:

- *Assignment Request*
An assignment request form is a guided activity that is used to request one or more role, privilege assignment for a given user. This form type is used for context based role and privilege assignments.
- *View Assignment Request*
A view assignment request form is a guided activity that is used to view the status of assignment requests. The form can also allow the user to restart failed or declined assignments.

- [Password Reset](#)

A password reset form is a guided activity that is used to enable users to reset their passwords from the Identity Management User Interface.

In addition to the normal form configuration, each of the guided forms has different parameters that specify the options and behavior available to the user performing the guided activity in the Identity Management User Interface.

2.5.15 Group Concepts

The Identity Center and the SAP provisioning framework support the following group concepts used in some repositories:

- A group defines the permissions the assigned users have in the back-end system and serves as a container object for all users needing the respective permissions (for example, Microsoft Active Directory).
- A group is just a set of users. Instead of assigning permissions (for example, roles) directly to users, administrators can assign permissions to group objects. The group members inherit the permissions assigned to the group automatically at runtime. The permissions are not directly, physically assigned to the users (for example, AS Java).

With the support of these group concepts, you can continue handling your group, role, and profile assignments in the repositories the way you used to. During the initial load, the group concept is applied automatically for the following repositories:

- ADS
- Sun One
- AS Java with database only
- AS Java with database and LDAP directory

Note

For AS ABAP repositories the provisioning of groups does not work, because the AS ABAP does not have a similar group concept. If the administrator tries, for example, to assign a group to a business role in AS ABAP, an error message appears.

Group Architecture

Group objects from repositories are represented in the Identity Center as a linked combination of an MX_GROUP object and an MX_PRIVILEGE object. This combines the advantages of the two entry types: Hierarchy support with the MX_GROUP and eventing support with the MX_PRIVILEGE. The SAP provisioning framework keeps these two objects synchronized. Whenever an assignment between any of the two objects and an MX_PERSON or MX_PRIVILEGE is changed, the same change is automatically applied to the other object.

Identities assigned to a group object in the repository are assigned to the MX_GROUP object and the MX_PRIVILEGE object.

Repository roles (which are represented as MX_PRIVILEGE objects) assigned to a repository group object are assigned to the MX_GROUP object. Note: These assignments cannot be applied to the group privilege as well, as it is not possible to assign privileges to other privileges.

Assignments between different repository groups from the same repository are represented as a link between the different MX_GROUP objects. Note: These assignments cannot be applied to the group privilege as well, as it is not possible to assign privileges to other privileges.

Consequently, the provisioning is triggered automatically as either the core part of the framework, an administrator or a job assigns the MX_PRIVILEGE (for the group object).

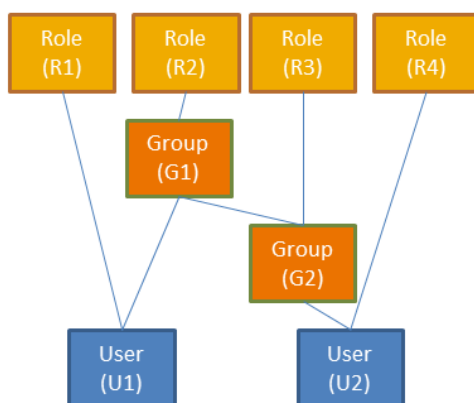
The following objects can be assigned to an MX_GROUP object:

- MX_GROUP
- MX_PERSON
- MX_PRIVILEGE

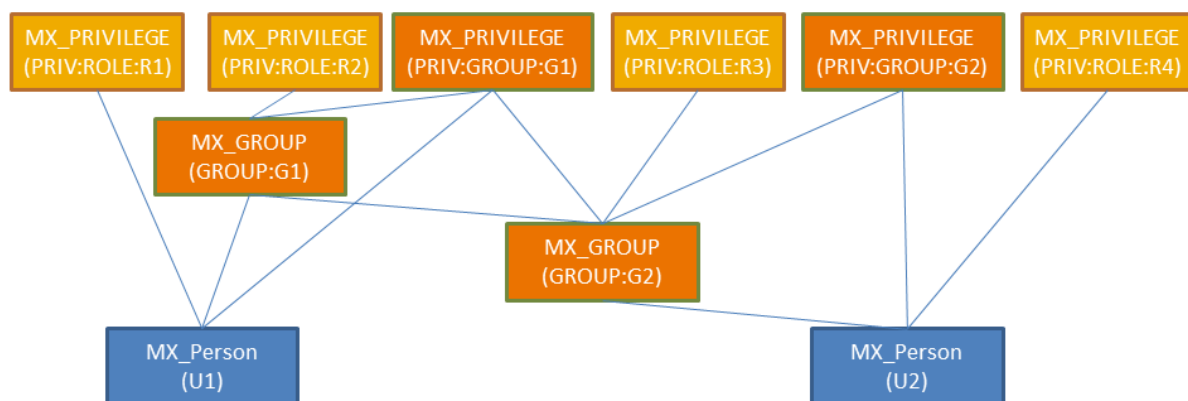
If all involved objects already exist in the back-end repository, the data is provisioned into the back-end system as it is.

As an example we take a closer look at an AS Java repository with DB-only configuration.

SAP NetWeaver AS Java



SAP NetWeaver ID Mmgt. 7.2



The Group (G1) of the AS Java is represented by the two objects MX_GROUP (GROUP:G1) and MX_PRIVILEGE (PROV: GROUP:G1) in the Identity Center.

On the AS Java, User (U1) is assigned to this Group (G1) and to the Role (R1) objects. Through the group assignment the user is also assigned Role (R2).

In Identity Management, the user is represented by the MX_PERSON (U1) object. The AS Java Role (R1) is represented by the MX_PRIVILEGE (PRIV:ROLE:R1) object and is directly assigned to the MX_PERSON (U1) object. The user's assignment to Group (G1) is represented in the Identity Management by two assignments: One to the MX_GROUP (GROUP:G1) assignment and one to the MX_PRIVILEGE (PRIV:GROUP:G1) assignment.

If in the Identity Center an MX_PERSON object is assigned to an MX_PRIVILEGE that represents a group, this assignment triggers the provisioning of the MX_PERSONMX_PRIVILEGE assignment to the repository. In addition, the core framework updates the assignment between the MX_PERSON and the corresponding MX_GROUP object.

On the AS Java there are groups from different datasources. If in the User Management Engine (UME) the datasource is a built-in groups adapter, then there is no user-to-group assignment in identity management. Such an assignment is not useful as the group membership of the users is determined at runtime. It depends on the status of the user, for example, whether he or she is authenticated or not. Identity Management does not display any members for the groups. If you add members to built-in UME groups using Identity Management and try to provision this group, UME displays an error message that the group is read-only.

For all other groups, Identity Management displays the group members and changes will be provisioned to the connected systems.

2.5.16 Identity Store

The Identity Center may contain one or more identity stores. The identity store is a central repository for managing identity information, such as employees, departments, roles, privileges and groups. The information in the identity store can then be used for provisioning and synchronization purposes.

One identity store is connected to the Identity Management User Interface in SAP NetWeaver AS for Java. If you need an Identity Management User Interface for another identity store, this requires a separate instance of SAP NetWeaver AS for Java. Each AS Java installation can only access one identity store. Therefore, to reduce the number of AS Java installations, we recommend that you set up one single identity store that is accessible from the user interface functions (for example, Workflow and Administration).

You can also use an identity store as temporary storage, for staging information before the entry is created in the productive identity store. You may have any number of identity stores for different purposes, for example for different organizations or for staging.

As opposed to a traditional database table, the identity store is dynamic, since attributes can be added as required. This makes it a very flexible solution.

The identity store is attribute based and offers extensive auditing functionality. Individual attributes can be modified, and all changes are tracked. All changes to each attribute can be kept for a given number of days or a given number of revisions.

Creating the Identity Store

When you create an identity store, a number of system attributes is added. They are used for different purposes in the system. For more information, see *Identity Store Schema*.

Entry Types

Entry types are used to define the properties of an entry in the identity store, such as which attributes are allowed and which attributes are mandatory.

Each entry within the identity store is identified by a unique identifier, MSKEY. This key is used to identify all attributes for the given entry. This number is unique across all identity stores in the same Identity Center.

The MSKEYVALUE

One of the attributes, MSKEYVALUE, is treated specially. This is the attribute that contains the unique identifier of the entry within the identity store across all entry types. This may be used to contain the employee number, the e-mail address or another value which uniquely identifies the user. This is also the user name for users logging on to the Identity Management User Interface and must match the UME user for authentication. This identifier is also used in provisioning on the entry.

For roles it is recommended to use `ROLE:<name>` and for privileges `PRIV:<name>` as MSKEYVALUE. For example `ROLE:Manager` or `PRIV:MyApplication`.

Note

It is not recommended to use any other attribute than MSKEYVALUE as the unique identifier in the identity store.

Deleting Entries

It is generally not recommended to delete entries in the identity store:

- Any event tasks must be able to handle deleted entries.
- Entries are needed for statistical and audit purposes.
- An employee may return to the organization at a later time, and it simplifies the process if the person entry can be re-used.

Instead it is recommended to set the entry as inactive or use a state attribute to indicate the state of the entry, for instance "Employed", "On leave" or "Not employed".

Referencing the Identity Store

From a job, the identity store is updated using a To identity store pass. The identity store can be used as a source in all To-passes instead of the temporary database by selecting the "Identity store" check box. In provisioning tasks, the identity store is always the source.

Related Information

[Identity Store Schema](#)

2.5.17 Identity Store Attributes

When you define an identity store, a number of entry types are included. For each entry type there is a number of predefined attributes. These entry types and attributes are described in *Identity Store Schema*.

Normally, which attributes are allowed are restricted to the predefined entry types and attributes, but you can also add attributes when you write entries to the identity store. Then you can select "Automatically create attributes" to add attributes as entries are written to the identity store.

If you are using entry types and "Automatically create attributes" is selected, adding new attributes to an entry of this type will also add these as legal attributes for this entry type.

Attributes starting with MX_ are reserved for use by SAP. Never start user defined attributes with MX_. Do not add or remove MX_ attributes from the built-in entry types (starting with MX_). Be careful if changing the built-in attributes starting with MX_, as this may have an effect on the system operation. Rather, it is recommended to create a custom attribute, as any changes will be overwritten with the next update of the schema.

Each attribute has a number of properties that defines how it is stored in the identity store (including any encryption of the attribute), how it is presented in the Identity Management User Interface or legal values. An entry's attributes can be updated by a task in the Identity Management User Interface or by a To identity store pass.

Related Information

[Identity Store Schema](#)

2.5.18 Identity Management User Interface

The Identity Management User Interface provides a web interface for initiating and approving provisioning tasks. The task structure and rules are defined using the provisioning module. The behavior of the task is

specified in the task properties. The configuration defines which users are able to execute which tasks and who is responsible for approving a given task. It also defines the list of attributes for each task.

As the functionality of the User Interface is highly configurable, it could be used in a number of ways. The following describes some typical scenarios.

The relevant tasks are presented to the user during log-on to the web interface. If the user is responsible for handling any approval requests, these are also presented in the web interface. The approver can then be notified of new approval requests by e-mail or SMS. A user may also be responsible for performing tasks on behalf of other users.

Tasks may also be initiated by an external system or based on events in the identity store. These tasks too may require approvals. This allows for distributed and delegated account management. New accounts are not automatically created based on additions in the HR system. The processing rules may require an approval from a manager before the actual provisioning of the account is done.

An approval may also include supplying additional information about the account, such as title or location. There may be several people involved in the process. For example, the mail system administrator may approve the creation of the mailbox for the user, by supplying information about the e-mail server on which to create the mailbox and other relevant information.

The User Interface can also be used for self-registration of accounts. New users create their own accounts by supplying the necessary information. When submitting this information, a task is initiated. The task configuration may include necessary approvals before the account is created.

For existing users, the User Interface can be used for self-service. The task configuration includes information about which attributes may be changed, and also which tasks to perform when one or more attributes have changed.

2.5.19 Inactive Entries

All entries in the identity store with entry type `MX_PERSON` or a custom entry type defined as an "Identity" entry type can be set as inactive. This has the same effect as deleting the entry:

- Assignments are removed and any related event tasks are executed
- The entry is no longer visible in searches, unless explicitly searching for inactive entries

Note

It is not possible to set entries as inactive if any of the options below is configured for an entry type:

- The *Identity Entry Type* option is not enabled.
- The *Context Entry Type* option is enabled.
- Relations to other entry types are configured on the *Relations* tab of a given entry type.

Note

It is not possible to modify an inactive entry, it can only be activated.

Setting an Entry Inactive

To set an entry inactive, set the attribute `MX_INACTIVE` to 1, either with a task in the Identity Management User Interface or in a To identity store pass.

Searching for Inactive Entries in the Identity Management User Interface

To be able to search for inactive entries in the User Interface, the search task referenced from the entry type must include the attribute `MX_INACTIVE`. When searching for entries, select "Advanced search", and select the "Inactive" check box in addition to any other search criteria. The search will return only inactive entries.

Setting an Entry as Active

To restore an entry, remove the `MX_INACTIVE` attribute, either by updating the entry with a task in the Identity Management User Interface or by setting the value to an empty string in a To identity store pass. Any event tasks will be run in the same way as when an entry is created.

2.5.20 Jobs and Passes

A job is a collection of passes, where each pass is responsible for performing one specific operation.

One pass will read data from one source repository, optionally process the data using scripts, and write to a target repository. A repository here also includes the identity store, as well as temporary database tables in the Identity Center database. Data processing tasks include cleansing and converting data, merging attributes into entries or performing other operations on the data.

Each job has a schedule rule that specifies when the job is to be executed. The dispatcher is responsible for starting the runtime engine for all jobs that are scheduled to be run. The schedule rule "On demand" is used to indicate that the job will only run when it is explicitly started, either from the Identity Management Developer Studio or the Identity Management Administration User Interface.

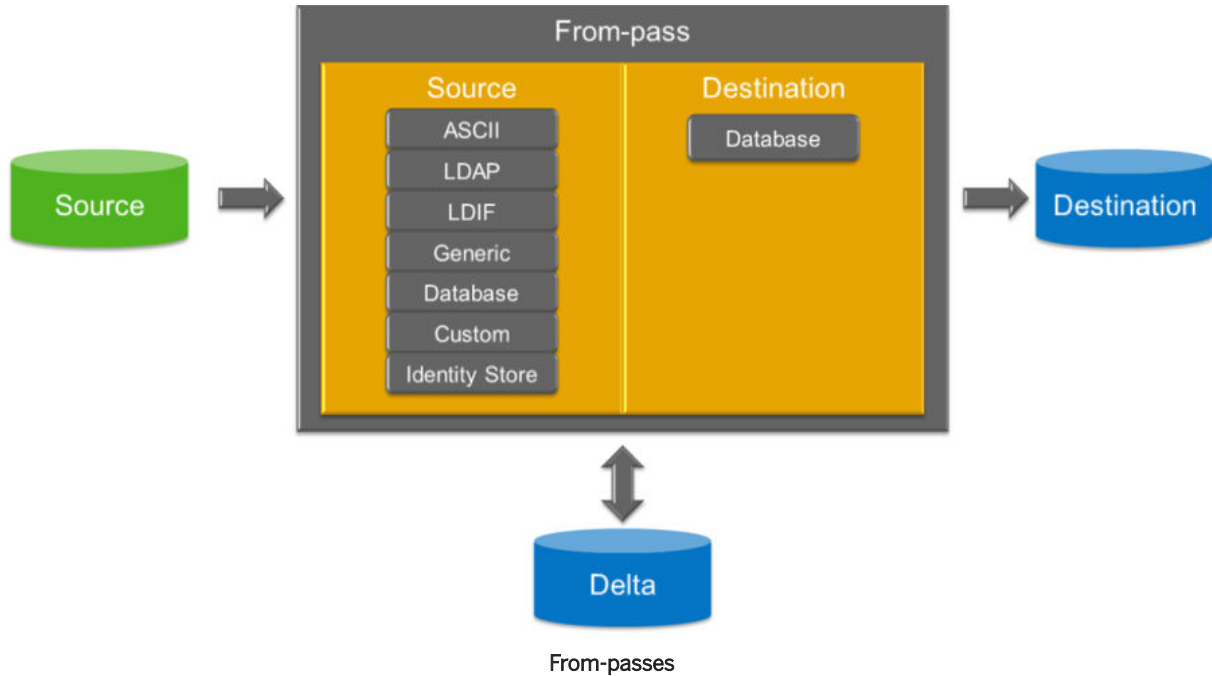
When developing a solution, a job can be run from the Identity Management Developer Studio. In a production environment, jobs are started by the dispatcher based on their schedule rule.

There are two main categories of passes:

- From-passes
- To-passes

From-Passes

From-passes are used to read from the data sources and have the temporary database as destination:

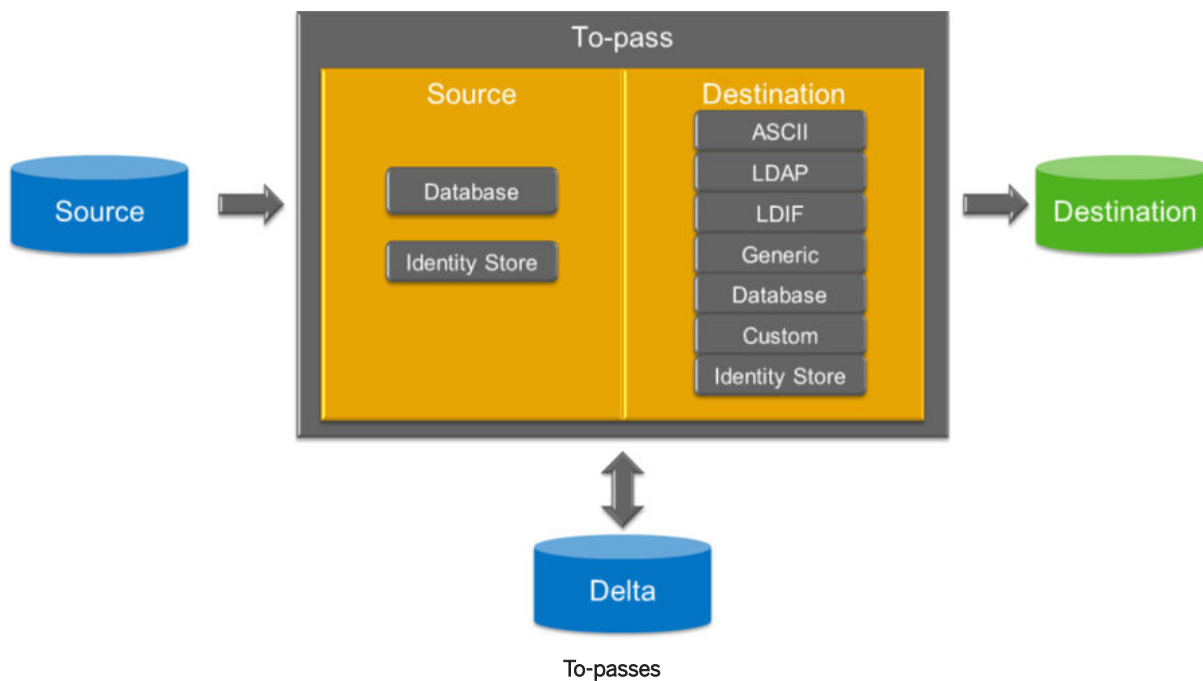


The source definitions of the pass specify how to read the data from the data source. The destination is always the temporary database.

Although normally not used in a From-pass, the delta mechanism ensures that only the modified data is processed.

To-Passes

To-passes are used to read from the temporary database table and have different types of data sources as destination:



The source of a To-pass can be the temporary database table that was updated by a From-pass, or the identity store.

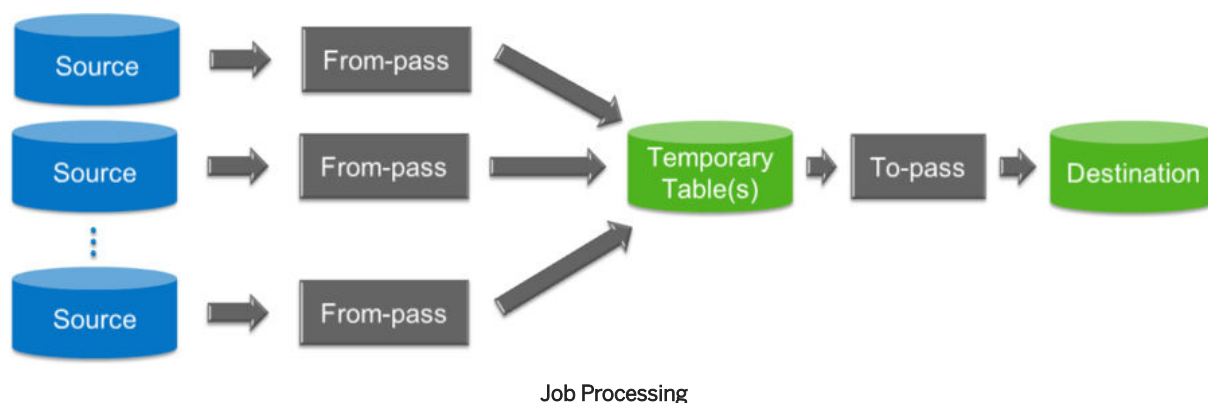
The definition of the destination specifies how the data is written to the target repository.

The delta mechanism has two purposes in To-passes:

- To reduce the write load (only changed entries are written).
- To detect deleted entries (i.e. entries which are no longer present in the source)

Running a Job

The runtime engine processes the passes of a job. There can be several From-passes reading from different repositories. The To-pass can then join attributes from several sources and create or update an entry in the target repository.



Related Information

[Repository Specific Job Execution \[page 105\]](#)

2.5.21 Member Event Handling

It is possible to specify processes that are executed when a privilege or a role is assigned to or removed from an entry.

When assigning a role or a privilege to a user, this assignment will have a status, and the assignment will not be active (and visible to other applications) until the resource defined by the privilege or resource is successfully assigned in the target system.

Note

For inactive entries, the event processes are executed if/when the entry is reactivated.

After the assignment request is submitted, a number of processes are executed, if defined. These processes are documented below.

Assignment event processes

It is possible to configure event processes which are executed when roles or privileges are assigned to users. The following event processes are defined:

- Validate process – Used to validate that an operation is allowed
- Execution process – Used to provision the resource to a target system
- Notification process – Used to notify about events

Common for all of these event processes is that a pending value object is created, and the process is executed on this object.

Validate Processes

These processes are only used to verify if the assignment/deassignment is allowed. These processes must never change any information, but only return a status indicating whether the operation is allowed or not. The

process can for example perform an approval, execute a request to GRC or check for membership in an Active Directory security group.

The typical use case for roles is to do an approval before the role is assigned, and for privileges to perform the add operation on the target system before the privilege is assigned. This means that if the role is rejected or the privilege execution fails, the assignment will not be performed. Also, the assignment will not be visible to the rest of the system until the task has completed.

If a validate process is defined, the corresponding execution process (Add Member, Remove Member or Modify Validity) will not be executed unless this validate process completes successfully.

The following processes are defined:

- **Validate Add** - This process is executed to validate that it is allowed to add the assignment.
- **Validate Remove** - This process is executed to validate that it is allowed to remove the assignment. This will only be executed when an assignment is removed explicitly, it will not be executed when the assignment expires as result of a Valid to setting or when the entry itself is deleted.
- **Validate Modify** - This process is executed to validate that it is allowed to modify the validity. The new values for Valid to and Valid from can be retrieved by using the syntax to retrieve properties from attributes in the identity store.

Execution processes

These processes are used to update the actual assignment by adding it, removing it or updating the validity. This process will typically connect to a target system, and grant or revoke a resource in that system. The execution processes are only called when the corresponding validate process has completed successfully, indicating that the assignment is allowed. If the assignment is not allowed, the corresponding execution process will not be executed. If no validate process is defined, the execution process is always allowed.

The following processes are defined:

- **Add** - This process is executed to add a resource to a user in a target system.
- **Remove** - This process is executed to remove a resource from a user in a target system.
- **Modify Validity** - This process is executed to update the validity of an existing resource assignment. Note that not all applications supports validity for assignments.

Notify Processes

The notify processes are used to notify about an upcoming event.

The following processes are defined:

- Link expiry notification

This process is typically called **<n>** days before the assignment expires. This process can for example send an e-mail to the user with a notification that the assignment is about to expire. Or it can automatically add 30 days to the valid to time for the link. In this case the Validate modify validity process will be executed, which may have an approval task for a manager to approve.

When to Execute the Process

For some process types, it is possible to configure when the process is to be executed. In any case, any defined validate processes will always be executed before the execution process.

The following options are available:

Value	Description
When Valid	For the Validate Add and Add Member processes this means on the Valid from time, and for the Link expiry process this is executed on the Valid to time.
Immediately	The process is executed when the assignment/deassignment is requested.
<i>Execute</i> field where to enter <n> days before	The process will be executed <n> days before the Valid from or Valid to time. You enter the <n> days in the <i>Execute</i> field.

Configuring Event Processes

It is recommended that the event processes are defined on the repository, and then overridden (if necessary) for a role or privilege.

Each role or privilege has one default repository. For a privilege, it is possible to configure a specific repository for the Validate processes, and for the Assignment and Deassignment processes. One reason to use a different repository for the Validate processes, is if you want to use GRC for validation. In this case, GRC will be the validation repository, while the Add and Remove member processes use the default repository.

Basic Event Processes

In addition to the already mentioned event processes, there are three basic event processes which can be defined on a privilege or repository. As opposed to the event processes already documented, these event processes execute on the user itself, and there is no pending value object involved. The following processes are defined:

- **Provision** - This process is executed after the link is set active, that is after the Validate add and the Add member processes have completed successfully.
- **Deprovision** - This process is executed after the link is removed, that is after the Validate remove and Remove member processes have completed successfully.

Modify Process

The Modify process is executed whenever an attribute on the user having a given privilege is modified. The Modify process can be defined on the repository and inherited to the privilege or defined directly on the privilege. The purpose of this process is to update any target repositories with information about the user, for example telephone number and e-mail address. If an attribute on the user is updated, the modify process for all connected privileges are executed.

It is recommended to define which attributes should trigger the modify process, to reduce the load on the system. This must be done on the [Task](#) tab in privilege properties.

The Modify process will be executed at least one time, but may be executed several times if more than one attribute is changed for a user. The same information could then be updated several times in the target systems. It is not possible to prevent multiple updates, so this must be handled in the target systems if it constitutes a problem.

If MX_DISABLED is defined as an attribute that triggers the Modify process, make sure you define it as the last trigger attribute in the list of modify attributes. Otherwise the Modify process for MX_DISABLED will be triggered multiple times, that will result in multiple email notifications, if Notification process is configured for this trigger attribute.

Request Complete Process

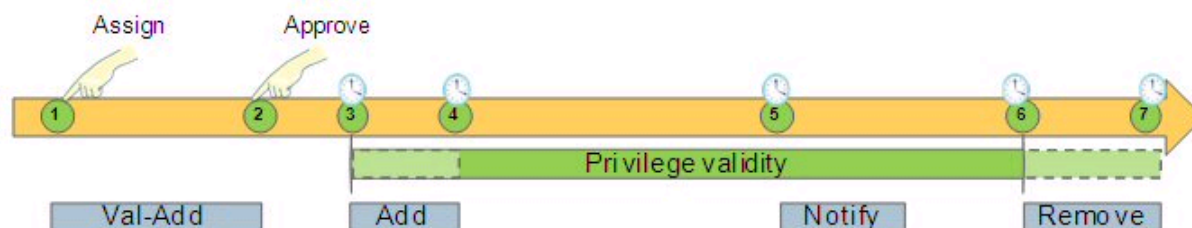
The Request complete process is executed when the last of a number of assignment requests with the same request ID is completed. This process can be used to provide a status back to an external system, for instance SAP BusinessObjects Access Control. You can define one Request complete process for each identity store.

Example

Normal assignment

The figure below illustrates an example of a typical configuration. In this example the timeline goes from left to right, the following processes are defined:

- **Validate Add** – Execute immediately
- **Add Member** – Execute when valid (i.e. on Valid from)
- **Remove Member** – Execute when valid (i.e. on Valid to)
- **Notify** – Execute N days before (i.e. N days before Valid to)



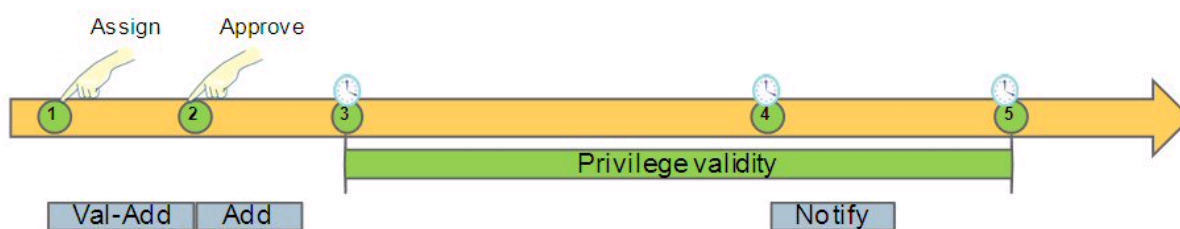
1. The user requests the privilege, with the given validity (ValidFrom=3, ValidTo=6). The Validate add process is started immediately.
2. The manager approves the assignment.
3. On Valid from the Add Member process is started, which will add the resource in the target system. This means that there may be a slight delay before the privilege is valid.
4. The Add Member process completes successfully, and at this time the privilege assignment is set as active.
5. N days before Valid to, the notify process is started, which sends an e-mail to the user that the privilege is about to expire.

6. On Valid from the Remove Member process starts, and removes the resource from the target system.
7. The Remove Member process completes OK, and the privilege assignment is no longer active.

ABAP privilege

Since ABAP "understands" validity, the process can be different. The following processes are defined:

- **Validate Add** – Execute immediately
- **Add** – Execute immediately
- **Notify** – Execute N days before (i.e. N days before valid-to)



1. The user requests the privilege, with the given validity (ValidFrom=3, ValidTo=5). The Validate Add process is started immediately.
2. The manager approves the assignment, and immediately the add process is started. This will write the assignment with validity to the target ABAP system.
3. On Valid from, the assignment becomes active within the Identity Center, but no action is taken. It is assumed that ABAP will do the same.
4. N days before Valid to, the notify process is started.
5. On ValidTo, the assignment becomes inactive, but no action is taken, as no del process is defined. It is assumed that ABAP will invalidate the assignment at this time.

Configuring Member Events

Member events for roles are configured as part of the role properties.

Member events for privileges can be configured both on the privilege and/or on the repository referenced from the privilege. Event processes are normally configured on the repository, but if there are some specific handling connected to the privilege, an event process can be configured on the privilege instead, which will override the settings on the repository.

--None/Inherited--

When configuring member events and event processes for privileges in SAP Identity Management User Interface, you can select the process itself or choose one of the following options:

Value	Description
None	No event process is triggered, regardless of whether the process is configured on the repository referenced from the privilege or on the repository type of the referenced repository.
Inherited	The event process configured on the repository referenced from the privilege is triggered. If no event process is configured on the repository, then the process configured on the repository type of the referenced repository is triggered.

Adding Master Privilege

If you configure member event handling for a repository, you can add a master privilege. This is a privilege on which other privileges for the same repository depend. This can be a privilege giving a user an account in the system referred to by the repository definition, which must be assigned before any access rights can be granted this account. Sub-privileges are privileges depending on the presence of the master privilege.

If there is a master privilege defined on the repository, any Add or Remove member process connected to the sub-privileges is not executed until the master privilege is assigned to the user.

Context Variables for Event Processes

For more information, see [Context Variables \[page 62\]](#)

Related Information

[Configuring the Event Tasks for a Privilege \[page 532\]](#)

2.5.22 Obsolescence

When a new version of a package is imported, jobs and tasks running at the time of the import and those being scheduled for processing in the provisioning queue before the package import, become obsolete. The same applies for scripts being called from jobs and tasks.

Overview

Obsolescence of jobs, tasks and scripts is introduced as of SAP Identity Management 8.0. It makes it possible to transport a package during the time of high system load without interrupting the provisioning and causing data inconsistency.

The following objects can become obsolete: jobs, tasks (action, switch, conditional, approval, attestation and UI tasks, that is to say forms), processes and package scripts.

When jobs and tasks of a given package version (1.0) are scheduled for processing, they must be processed based on the job and task definitions of this package version (1.0). When a new package version (1.1) is imported, the tasks and jobs will be processed based on their new definitions. If there are still some jobs and tasks of the old package version (1.0) that were scheduled before the package import, they will be processed based on their old definitions. After the package import, there might be a case when in the provisioning queue there are one and the same jobs and tasks belonging to different package versions.

Obsolete objects are not visible and searchable in the Identity Management Developer Studio. Therefore, if an error occurs while an obsolete job is running, it is not possible to select it and manually restart it.

→ Recommendation

If an obsolete job goes into an error state, proceed as follows:

Create a job that executes the `mxi_repair_obsoleted_jobs` stored procedure to change the state of obsolete jobs in the provisioning queue from `Error` to `Idle`. The dispatcher will be able to run those jobs again.

Mechanism

The obsolescence mechanism works as follows:

- During package import both versions of an object are stored in database tables. The new version of the object is inserted in the database as a new record. The record for the old version of the object is modified. The process of modifying the old version of an object in the database tables is called obsolescence.
- Each import of a new package version generates an obsolete version of the obsolete package objects. This means that in the database there might be multiple versions of the same object.

ⓘ Note

There will always be only one current version of an object and one or several obsolete versions of this object, depending on the number of new package version imports.

Note

Each time a new version of a package is imported, the new versions of the jobs and tasks are stored in the database tables as new insert records with new IDs. During package transport all task references from other objects, such as: tasks, forms, entries, attributes and package constants, are updated to point to these new IDs.

The following database tables and columns are affected by obsolescence:

Table	Column	Value
mxp_tasks	description	Obsoleted - this value is set for obsolete tasks.
	mcObsoletedGUID	Possible values: • <GUID> - the GUID this task had before it became obsolete. • NULL - this value is set for non-obsolete tasks.
	mcObsoletedBy	Possible values: • <taskId> - taskId reference of the task that replaced this task. • NULL - this value is set for non-obsolete tasks.
mc_jobs	description	Obsoleted - this value is set for obsolete jobs.
	mcObsoletedTime	Possible values: • <timestamp> - timestamp of the time the job became obsolete. • NULL - this value is set for non-obsolete jobs.
mc_package_scripts	mcIsObsoleted	Indicates whether a script is obsolete or not. Possible values: • 0 - Indicates that the package script is not obsolete. • 1 - Indicates that the package script is obsolete.

2.5.23 Parallel Provisioning

Normally, an action task is single-thread to prevent problems that could occur if a task was to process several entries to the same repository. By default, the action task is also single-thread when processing entries to different repositories. This means that provisioning to one repository has to wait for processing to another repository to complete.

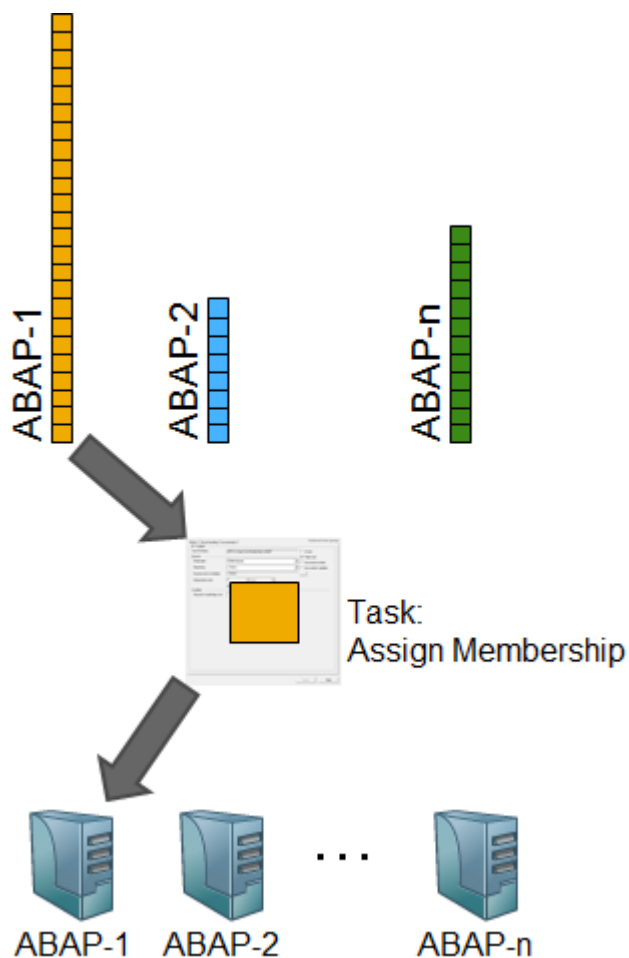
In situations where you have a large number of repositories of the same type, and many entries to be processed to several repositories at the same time, parallel provisioning improves performance by allowing provisioning to different repositories to run in parallel. Provisioning to one repository still is done in sequence.

Parallel provisioning must be enabled on the action task. Virtual copies of the action task are created dynamically for each new repository, which can be viewed on the action task's job.

The following diagrams illustrate parallel provisioning:

- **Parallel Provisioning is not enabled**

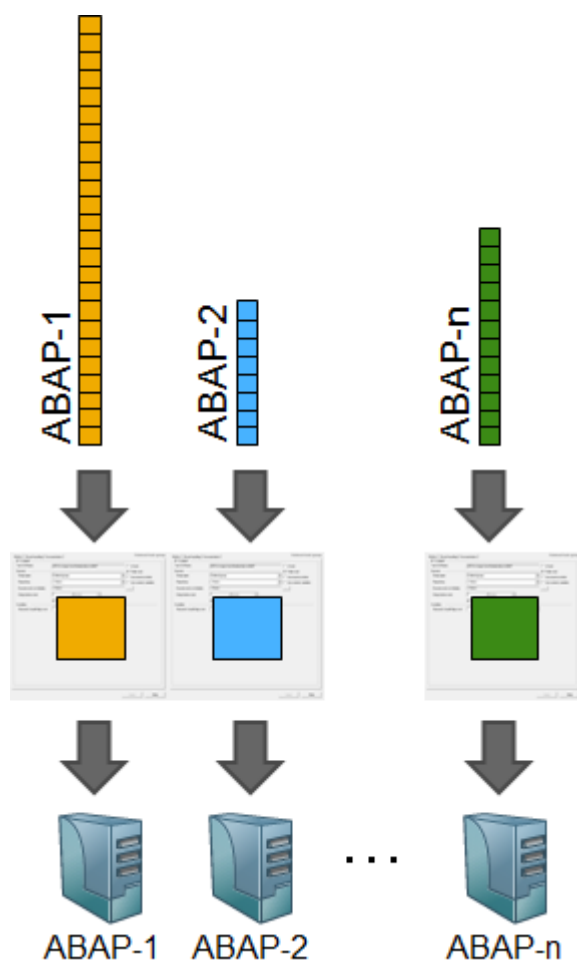
This illustrates the situation without parallel provisioning enabled. There are many entries in the queue to be processed for each repository, but all processing must be done with the single-thread task.



- **Parallel Provisioning is enabled**

This is the situation when parallel provisioning is enabled. Virtual copies of the job are created for each repository, which means that entries for each repository can be processed in parallel. It is not necessary

to specify which repositories are being used in advance, as these are added dynamically when entries are processed.



Considerations

There are some considerations/restrictions that apply when using parallel provisioning, including:

- Delta must not be enabled for the job. There is only one delta key, meaning that it is not possible to distinguish the delta for each repository. Delta is normally not enabled for an action task's job.
- Package variables will be shared for all instances of the job.
- If the job writes to an ASCII file, the same ASCII file will be updated from all instances of a job.

Related Information

[Action Task \[page 1129\]](#)

[Viewing and Configuring Action Task Properties \[page 239\]](#)

2.5.24 Packages

A package is a collection of configuration information, such as constants, scripts, repository types, processes, forms and jobs.

A package is the smallest part of the configuration that is maintained as a unit. This could be a connector for a repository type or a collection of utilities that are used by other packages. Users are granted access to different packages which allows multiple users to work on the configuration and transport separately. The configuration packages are delivered as part of the Identity Management core component and imported to the Identity Management database to provide a starting point for the configuration of the solution.

Qualified Name

Each package has a package qualified name, consisting of a base qualified name provided during the installation and configuration of SAP Identity Management and a package name given to the package when it is created.

ⓘ Note

The base qualified name can contain only alphanumeric values (A-Z, a-z, 0-9), underscore (_) and period (.). Following the convention, you can use your company's reversed Internet domain name for the base qualified name of your packages. For example, if your company has the domain name **acme.com**, you can use the reversed domain name **com.acme** for the base qualified name.

ⓘ Note

The package name should be globally unique. Globally means that you cannot have the same package even in different identity stores.

Example

Base qualified name: com.acme

Package name: MyPackage

Package qualified name: com.acme.MyPackage

User Authorizations

Users must be authorized to get access to the content of a package. A user may have different authorizations in different packages. The following package authorizations exist:

- View

- Developer
- Layout Developer
- Import
- Owner

For more information, see: [Authorized Users \(Packages\)](#).

Multi-User Editing

The package must be checked out in order for the configuration to be modified, and then checked in to make the updated configuration available to other packages. No other user can work on the same package once it is checked out, while users can work on different packages (different parts of the configuration) at the same time without overwriting each-others' changes.

Versioning and Revision Control

The package contains two version numbers, a major version and a minor version. The major number is incremented every time the public interface is changed (for example, a process is made public). The minor number is incremented every time the package is checked in, regardless of whether the package has been modified. Adding cross-package references to public objects increments the minor version of the package.

A previous version of a package can be restored and unsaved (not checked-in) changes reverted. For more information, see: [Revision History \(Packages\)](#) and [Restoring a Version of a Package](#).

Transport

Transport of configuration is done on a package level. Each package is transported separately.

Public Objects

Objects within a package can be defined as public, while others are private to the package. A public object means that it is available to be called by other packages. Processes, scripts and forms can be public in a package.

Related Information

[Authorized Users \(Packages\) \[page 1009\]](#)

[Restoring Version of a Package \[page 198\]](#)

2.5.25 Package Constants

Package constants are only available within the package. You can call them from jobs, tasks and scripts within the package.

If you want them to be available for jobs in other packages, you need to create your custom package script and call the package constant from this script using the following syntax: `%%$pck.<constant>%`. Additionally, you can use the function [uGetConstant](#) to retrieve the value of the given constant.

Retrieving a Package Constant from Another Package

❖ Example

In package 1, you have created a PHONE_PREFIX package constant of type String with value +49. You want to make it available for a job in package 2. Proceed as follows:

1. In package 1, create a custom package script to call the `%%$pck.PHONE_PREFIX%` package constant.
2. Make the custom package script public.
3. In package 2, open the job, select the [Scripts](#) tab and choose [Add Link to Package Script](#) from package 1.
4. Run the job.

Retrieving a Package Constant Within a Package

❖ Example

In package 2, you have created a CELL_PHONE package constant of type String with value 12345. You want to make it available for the job in the same package 2. Proceed as follows:

1. In package 2, create a custom package script to call the `%%$pck.CELL_PHONE%` package constant within the same package.
2. In package 2, open the job, select the [Scripts](#) tab and choose [Relink to Package Script](#) from the same package 2.
3. Run the job.

You can also run one job to retrieve both -a package constant from another package and a package constant within a package. In this case, you need to create two separate custom package scripts, one for each package.

Note

When you use a script to retrieve a package constant, this constant will first be searched for in the package from which the script is invoked (package 2). If the constant is not defined in this package, then it will be searched for in the package which contains the script (package 1).

Note

If you want a given script to retrieve a package constant which name is dynamically generated (for example, by concatenating "pck." + "PHONE_PREFIX"), you need to use the [uGetConstant](#) function. The script will not be able to retrieve a package constant with the following syntax "%\$pck." + "PHONE_PREFIX" + "%".

A package constant that is delivered with an SAP Provisioning framework package can be used in other packages in a similar way. For example, the engine package contains the `;%$pck.SAP_DELIMITER%` package constant. To make it available for a job in AS Java connector package, in the AS Java connector package you need to add link to the `getMultiValueDelimiter()` package script from the Engine package.

Package constants are listed in the context menu and can be used in any field in the pass properties. Package constants can be edited in SAP Identity Management Administration User Interface, if [Show in the Admin User Interface](#) checkbox is selected when you create the package constant.

Note

When upgrading from SAP Identity Management 7.2 to 8.0, global constants are replaced by package constants. Nevertheless, constants can still be referred to with `;%$glb.%` prefix due to backward compatibility. Scripts that reference package constants with `;%$glb.%` prefix will continue to work, but you will get a warning in the Job Log. The syntax of package constants is kept the same, only the prefix has changed.

For example:

In version 7.2: `;%$glb.PHONE_PREFIX%`

In version 8.0: `;%$pck.PHONE_PREFIX%`

Related Information

[Adding Package Constants \[page 208\]](#)

[uGetConstant \[page 1160\]](#)

[Modifying Package Constants](#)

2.5.26 Pending Value Objects

A pending value object is an entry with entry type MX_PENDING_VALUE. Pending value objects are created in the following cases:

- When executing member event processes.
- For future attribute values for regular attributes (not reference attributes).

Member Event Processes

A pending value object is created when executing a member event process for an assignment (role, privilege etc). The member event process is executed on the pending value object. In this case the pending value object contains the reference identifier in the attribute MX_LINK_REFERENCE.

The status of the assignment is updated (for instance, the role is assigned) when the process has completed. This depends on the value specified for *Wait for event tasks* on the *Result handling* tab of the process.

The member event processes can read attributes from all involved entries of the pending value object.

Some time after the process has completed, the pending value object will be deleted.

For details about the attributes of the pending value object (MX_PENDING_VALUE entry type), see the document *Identity Store Schema*.

Related Information

[Identity Store Schema](#)

2.5.27 Performance Optimization

The following properties and provisioning constants can be set for an identity center to control the behavior, which can improve the performance:

- Modify policy of privileges
- Modify interval of privileges
- MX_DIRECT_PRIVILEGE (Direct privilege inheritance)

Modify Event Handling

The modify policy of privileges is a property of the identity center and specifies how to handle modify events on privileges. Its value can be changed in the Identity Center properties. Modify events on all other entry types are not affected.

The default behavior is that the modify event process is executed for all attributes of the privilege. You can also select a number of attributes that should trigger the modify event process.

With this property, you can specify the following behavior:

- **0** (or missing)
Use the default behavior (the modify event process is executed for all attributes or for the attributes specified with `MX_MODIFYTASK_ATTR`).
- **1**
Do not execute the modify process for privileges. This will improve performance, as all checking for modify events are skipped, and can be used if you do not rely on modify events.
- **3**
Modify event processes are only triggered if the attribute is listed by `MX_MODIFYTASK_ATTR`.
If there is no `MX_MODIFYTASK_ATTR`, the modify process will not be started.

Note

This behavior is the reverse of the modify policy property = **0**, where the lack of `MX_MODIFYTASK_ATTR` means that every attribute will trigger the modify process.

Using this strategy will also improve performance, compared to **0**.

Modify Event Interval

The modify interval of privileges is a property of the identity center and is used in the `mx_i_privilegeevents` stored procedure. It holds an interval in seconds, which is used to determine whether an attribute modification triggers provisioning. The default value is **10** seconds. It can be changed in the Identity Center properties.

Direct Privilege Inheritance

The package constant `MX_DIRECT_PRIVILEGE` specifies whether direct privilege inheritance is used in the configuration or not.

Note

If not used, the package constant should be set to `FALSE` for better performance.

If the configuration does not use direct privilege inheritance, the constant may not be visible (default value is `FALSE`).

The constant can have the following values:

- **TRUE**
The configuration uses direct privilege inheritance.
- **FALSE**
The configuration does not use direct privilege inheritance. Recommended value for performance optimization if not needed. This is the default value for new installations.

2.5.28 Privileges

A privilege entry type defines a privilege to a given resource, for instance an access to a given system.

Privileges are used to manage provisioning based on assignments of authorizations in external systems. These may be user accounts, e-mail accounts or access control systems. The fact that a user is given a privilege does not in itself mean that the actual account is created, only that the process has been initiated. Assigning and removing privileges can automatically start tasks to perform provisioning and de-provisioning.

Privileges can be assigned to a user in two ways:

- Directly, by adding the privilege to the user. Assigning direct privileges is done by adding the user entry as a member of the privilege entry.
- Indirectly, through a role. When assigning a role to the user, all privileges associated with the role (and all sub-roles) will be added to the user.

When a role is removed from a user and an indirectly assigned privilege is removed along with the role, all other roles assigned to this user must be checked to see whether they reference this privilege. Only privileges that are not assigned through other roles can be removed.

Privilege Dependencies

Typically within one repository, there is one privilege which is used to create an account within the target application, and other privileges which are used to grant various access rights to that account. The account must be created before any access rights are granted. The privilege dependencies is a mechanism that guarantees that the account will be created before the access rights are given an entry. The following two terms are of importance:

- Master privilege: This refers to any privilege on which other privileges depend, e.g. an account privilege.
- Sub-privilege: This refers to any privilege which depends on the presence of another privilege.

Privileges that are used in the SAP Provisioning Framework are:

Account Privileges

The account privilege is created by the initial load job, with the following naming convention:

PRIV:<Repository>:ONLY

The privilege has the attribute `MX_IS_ACCOUNT` that indicates that this is an account privilege, which means that any operations on the repository depend on the presence of this privilege on the user. During the initial load, this privilege is assigned to the users read from the repository (prior to assignment of authorization privileges). The account privilege is not visible in the Identity Management User Interface while the initial load job is running.

The account privilege is used for users or company addresses:

- Assign it to a user when creating the user. This assignment indicates that changes to the user are to be provisioned to the corresponding system. To delete the user in the corresponding system, you remove this privilege.

Note

The account privilege cannot be assigned with context to a user. That is, when implementing context-based assignments functionality, you cannot add context to an account privilege assignment.

- Assign it to a company address. This assignment indicates that changes to the company address are to be provisioned to the corresponding system. To delete the company address in the corresponding system, you remove this privilege.

System Privileges

The system privilege is created as part of the repository definition with the following naming convention:

`PRIV:SYSTEM:<REP_NAME>`

The system privilege must be unique for each repository and should never be changed. It is automatically assigned to each user or group when the assignment of the account privilege is done. It is handled internally and should never be assigned directly to any user.

Function Set Privileges

When working with the enhanced SAP Business Suite use case, function set privileges are created for each of the SAP Business Suite applications that are activated for provisioning with SAP Identity Management. These privileges have the following syntax:

`PRIV:FUNCTION_SET:<Repository>:<BApI_Filter_Value>`

Assign this privilege to business roles or identities that use the corresponding SAP Business Suite application.

2.5.29 Provisioning

Provisioning is the act of providing something for use, normally user and account provisioning.

In the case of the Identity Center, this normally means creating objects and setting attributes within various repositories. This can, for example, be creating users in an LDAP directory or a database, enabling certain

operations by setting attributes within the repositories etc. In the Identity Center, provisioning is based on setting up a number of tasks to perform the given actions.

The provisioning module of the Identity Center offers a provisioning system that has the following characteristics and features:

- *Persistent and state driven*
For a provisioning system, persistence is important. In real life, servers will fail, networks go down or there will be power glitches. The use of a database to keep a persistent state for all operations will guarantee that no information is lost.
- *Event driven*
By using event tasks on privileges, provisioning is automated based on the assignment of the privilege.
- *Auditing*
It is important to be able to go back, and find out who was allowed to do what, at which time. There are legal requirements in this area, including HIPAA (Health Insurance Portability and Accountability Act) and Sarbanes-Oxley.
- *Chaining of tasks*
In many cases, there is a need for doing a number of tasks in sequence (or parallel).
- *Delays between tasks*
There are cases where you want to set up a delay between two tasks. For example when a user leaves the company, the user account will immediately be disabled, and after 2 weeks, the account will be removed.
- *Retries*
Retries can be defined for all tasks. The retries will take effect when errors occur.
- *"Wait for"*
This means that a task can wait for a certain condition (for example an audit flag) to become true, or to wait for all event tasks to finish before continuing.
- *Rollback*
Rollback can be defined across several repositories, so that during a failure, it is possible to undo operations in all involved repositories.
- *Context variables*
Context variables contain context specific information, such as which attributes were updated etc. Can be used to control task execution.

2.5.30 Reconciliation

Performing structural changes to roles and privileges that affect multiple entries may be very time consuming and process intensive. One example is a change to a role hierarchy with a large number of roles and thousands of users, which all are assigned inherited roles. If one of the branches in the role hierarchy is disconnected, the roles and privileges of all of the thousands of users must be recalculated, which may put a very high load on the system for a long time if it was to be a synchronous operation performed in a single transaction.

The "Dirty" Flag

For structural changes in the identity store, the affected entries are marked as "dirty" to indicate that there are changes that must be processed at a later time. The "dirty" flag is set on the affected entries in the following cases:

- Structural changes in the role tree (roles/privileges).
- When changing properties of a role or privilege like Allowed for/Not allowed for, Autoassign, Inheritance direction.
- Changes to context properties on privileges and roles (except "Context type").
- An entry can also be set "dirty" explicitly by using the function `uIS_SetDirty`.

The "dirty" flag is not set in the following cases:

- When changing properties of a repository.
- When changing the context type configuration for a role or privilege.

To be able to reconcile the entries after doing such changes, you must identify which entries are affected and use the function `uIS_SetDirty` to set the "dirty" flag.

Reconciliation

Reconciliation of "dirty" entries is done regularly with a scheduled procedure for housekeeping executed by the dispatcher.

When executed, the Reconcile dirty entries procedure can do the following:

- Add all missing assignments of the given entry.
- Delete all additional (not needed) assignments of the given entry.
- Retry all failed assignments of the given entry.

Note

Keep in mind that the assignments of a user are processed one by one - this means that if you need to add three new privileges to a user, you can expect three separate provisioning processes to be triggered.

A manual reconciliation can be initiated by using the internal function `uIS_PrivReconcile`. This function has the `MSKEY` as a parameter, and will perform any reconciliation for this entry.

If a reconciliation of the entry is not sufficient to correct the assignments of an entry, the entry can be repaired by calling the function `uIS_RepairEntry`.

Related Information

[uIS_PrivReconcile \[page 1184\]](#)

[uIS_RepairEntry \[page 1185\]](#)

2.5.31 Repository Types, Constants and Variables

For all the systems you want to connect to SAP Identity Management, you need to create repositories based on repository types. Repositories are configured with a set of constants, both repository and repository type specific.

Repository Types

A repository type defines common parameters (constants) for all repositories of the given repository type. This brings the following advantages to the repository configuration process:

- You can change a repository constant once for all repositories of a given type. This applies to both already created repositories and new ones.
- You can add a new repository constant to all repositories of a given type. This applies to both already created repositories and new ones.

All Constants Declared in the Type

All constants are declared in the repository type. They are defined by name, category, data type, value and (optionally) description. All repository constants declared in a repository type are stored in `mc_repository_type_vars` database table. Repository constants defined for a specific repository are stored in the `mc_repository_vars` database table only.

For more information about the view returning the repository constants, see [Views for Repository Constants \[page 153\]](#).

For more information about the different categories of repository constants, how they are handled in the user interface and how they are represented in the database, see *Constants* in [General \(Repository Type\) \[page 1012\]](#).

Identified by Qualified Name

A repository type is identified by a qualified name. For example: `com.sap.idm.connector.asjava:reptype.ASJavaDB`, where `com.sap.idm.connector.asjava` is the package qualified name and `ASJavaDB` is the repository type name. A package may contain one or more repository types. Repository types are always public and are only referenced from repositories.

Creating, modifying and deleting repository types is done in the Identity Management Developer Studio. Deleting a repository type or a package results in orphaning the repositories. Orphaned repositories cannot be modified and repository jobs for orphaned repositories cannot be run. Adding the repository type or reimporting the deleted package can reconnect the orphaned repositories to the repository type again. See [Deleting a Repository Type \[page 217\]](#)

Changing the repository type of a given repository is done in the Identity Management Administration User Interface. See [Changing Repository Type of Repositories \[page 499\]](#).

Package Transport

During package transport, modified values of repository constants overwrite the existing ones with the following exception: constants of types other than `Job Reference` and `Process Reference` that have [Show in the Admin User Interface](#) enabled. See [Importing Packages \[page 201\]](#)

Repository Constants

When defining a repository definition using a template, the specified values are stored as repository constants (or variables). The values for the repository constants are used to access the data source and they are available from the context menu in the passes that reference this repository definition, or any pass that inherits a repository definition from one of its parent tasks.

Repository constants are called using the following syntax:

```
%rep.<constant>%
```

From scripts, the repository constants can only be referenced by using [uGetConstant](#) function.

General Remarks

Below, you can find general remarks on repository constants for repository definitions used by SAP provisioning framework.

- Repository constants can be both repository and repository type specific. Constants that are repository specific are configured in the Identity Management Administration User Interface (Admin UI). Constants that are repository type specific are not visible and therefore not configurable in the Identity Management Admin UI, except for repository type constants with override. See [General \(Repository Type\) \[page 1012\]](#)
- To adapt connectivity constants that define the connection to the backend system, you adjust them directly without having to delete and re-create the repository.
- You should not change any of the following constants:
 - MX_PRIV_GROUPING_ATTRIBUTE - Property attribute for grouping.
 - MX_GROUPING_RULE

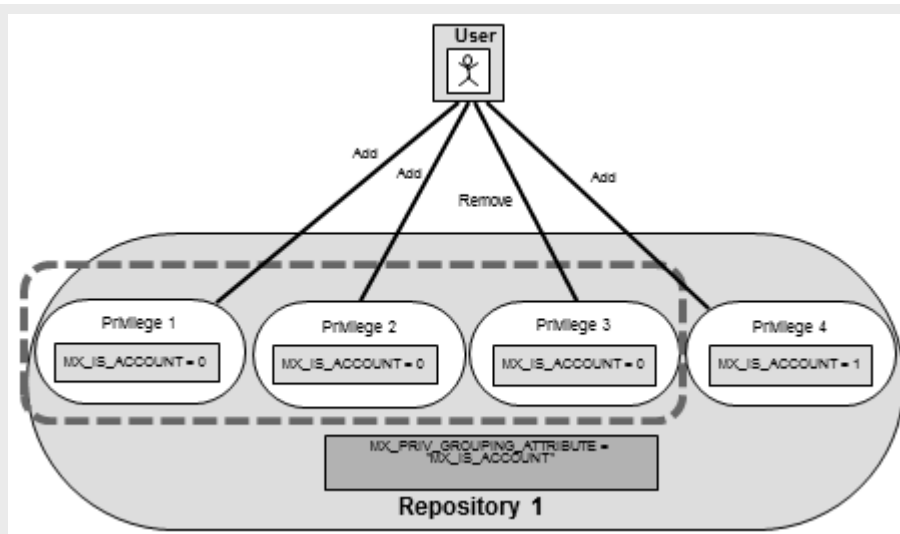
The repository templates use the following values:

Values for MX_GROUPING_RULE

Value	Comment
P:4	With this value, the system groups privilege assignments if the assigned privileges have the same value for the attribute defined in MX_PRIV_GROUPING_ATTRIBUTE constant of their repository.
P:5	Same as P:4 but the operation type, for example, remove or add is also taken into account. Privilege assignments are only grouped if the operation type is also the same.

Example

Example for Privilege Grouping in AS ABAP Repositories



P:4

The MX_PRIV_GROUPING_ATTRIBUTE repository constant has the value MX_IS_ACCOUNT. MX_IS_ACCOUNT is an attribute defined at the privilege. It has the value 0 for all privileges representing ABAP roles or profiles, while the system-specific privileges **PRIV:SYSTEM:<Repository>** and **PRIV:<Repository>:ONLY** have another value. Therefore, all assignments of ABAP roles and profiles are grouped and thus are processed in a single provisioning. This applies to privileges 1, 2, and 3. The different operation types shown in the figure above are not relevant with this grouping rule. If the attribute has a different value the privilege is not grouped (privilege 4).

P:5 (not included in the figure above).

In addition to the criteria valid for P:4, the operation type for the privilege is also taken into account. Privilege 3 would not be grouped together with privileges 1 and 2 because it has a remove operation.

- REPOSITORY_SYNC
 - REPOSITORY_TYPE - Describes what kind of system is connected. In the templates the values are, for example, ABAP, DUALABAP, JAVA, LDAP, and SUN.
- For more information about repository constants used for each repository type, see [Repository Constants \[page 1234\]](#).

Repository Type Constants

Below, you can find the available repository type constants in SAP Identity Management:

Repository Type Constants

Constant Name	Description/Values	P(privi- lege)/R(role)
MX_ADD_MEMBER_TASK	See Privilege and Role Assignment Attributes	P/R
MX_DEL_MEMBER_TASK	See Privilege and Role Assignment Attributes	P/R
MX_DEPROVISIONTASK	See Privilege and Role Assignment Attributes	P

Constant Name	Description/Values	P(privi- lege)/R(role)
MX_LINK_EXPIRY_NOTIFICATION	See Privilege and Role Assignment Attributes	P/R
MX_MOD_VALIDITY_TASK	See Privilege and Role Assignment Attributes	P/R
MX_MODIFYTASK	See Privilege and Role Assignment Attributes	P/R
MX_OFFSET_ADD_MEMBER	See Privilege and Role Assignment Attributes	P/R
MX_OFFSET_LINK_EXPIRY	See Privilege and Role Assignment Attributes	P/R
MX_OFFSET_VALIDATE_ADD	See Privilege and Role Assignment Attributes	P/R
MX_CTX_AUTO_STRATEGY	See Assignment Context Attributes	P
MX_PRIV_GROUPING_RULE	Used to define the grouping for the privileges for given repository definition. Legal values: - 0: No grouping - P:0 All grouped in one request - P:1 Grouped by operation - P:2 Grouped by system - P:3 Grouped by system and operation - P:4 Grouped by privilege property only - P:5 Grouped by privilege property and operation - P:6 Grouped by privilege property and system - P:7 Grouped by privilege property, system and operation See Grouping of Privileges [page 744].	-
MX_PROVISIONTASK	See Privilege and Role Assignment Attributes	P/R
MX_REQ_PRIV	See Privilege Dependencies Attributes	P
MX_REQ_PRIV_NOMASTER_TASK	See Privilege Dependencies Attributes	
MX_REQ_PRIV_INTERVAL	Not in use (as of SAP NetWeaver Identity Management 7.2 SP9).	P
MX_REQ_PRIV_PCYADD_MISSING	Not in use (as of SAP NetWeaver Identity Management 7.2 SP9).	P
MX_REQ_PRIV_PCYADD_PENDING	Not in use (as of SAP NetWeaver Identity Management 7.2 SP9).	P
MX_REQ_PRIV_PCYADD_REMOVING	Not in use (as of SAP NetWeaver Identity Management 7.2 SP9).	P
MX_REQ_PRIV_TIMEOUT	See Privilege Dependencies Attributes	P
MX_VALIDATE_ADD_TASK	See Privilege and Role Assignment Attributes	P/R
MX_VALIDATE_DEL_TASK	See Privilege and Role Assignment Attributes	P/R
MX_VALIDATE_MOD_VALIDITY_TASK	See Privilege and Role Assignment Attributes	P/R

Constant Name	Description/Values	P(privi- lege)/R(role)
REPOSITORY_TYPE	String value indicating the type of repository definition. Value examples: LDAP, ABAP, SUN, JAVA, DUALABAP, GRC, NOTIFICATION.	-

Repository Variables

A typical use of a repository variable is to keep track of how much data is read from a source repository, i.e. an LDAP change number or a database timestamp. When reading data from the repository, the job will then "remember" how much was read in the last execution, and read the newer entries, while updating the repository variable with the new timestamp.

Repository variables are available only from scripts. To modify or create a repository variable, use the `uSetRepositoryVar` internal function. To get a repository variable, use the `uGetRepositoryVar` internal function. The variables are visible only in the `mc_repository_vars` table in the database.

2.5.32 Repository Specific Job Exection

If you have a number of repositories of the same type, for instance a number of ABAP servers, processes like initial load or reconcile could be handled by the same job. This is more efficient than creating a job for each repository that has to be duplicated/maintained. A job can be executed with the repository as a parameter. In this case you must specify a repository type as part of the job properties, and pass a specific repository as a parameter when running the job.

In a system landscape with a QA and production environment where you transport the configuration from QA to production, the repository type is transported to the target environment and you can create repositories of the same type from the Identity Management Administration Interface.

Related Information

[Jobs and Passes \[page 78\]](#)

2.5.33 RequestID

The request ID is a general purpose request identifier used to group several assignments/deassignments into one request. The request ID can be used for instance when integrating with SAP BusinessObjects Access Control.

The criteria for the request are:

- All assignment/deassignment requests must concern one user.
- All assignment/deassignment requests must be submitted within one operation, for instance by one "submit" from the Identity Management User Interface.

The request ID is added when a assignment/deassignment request is initiated either from the Identity Management or by a job. From a job, you use the specified syntax in the To identity store pass to write the assignment with a request ID.

The Syntax of the RequestID

The request ID is a global value, so for the Request complete task to be able to identify which process did the assignment, the following syntax is used:

PREFIX#ID

PREFIX identifies the process that did the assignment.

Note

The PREFIX cannot start with MX, as that is reserved for internal use.

ID can be any string, but must be unique within one PREFIX.

The Request Complete Task

The request ID is used by the Request complete task for an identity store. This task is executed when the last of the event tasks for assignments/deassignments with the same request ID is completed. This also includes processing of all inherited assignments (including any approvals), regardless of whether the tasks completed successfully or not. When the Request complete task has been executed, no more processing will be done on the request.

There is a sample task that you can import and use as a basis. The sample is located in the folder `\Templates\Identity Center\SAP Provisioning framework\RequestComplete.mcc` in the Identity Center installation. Import this task to a suitable node in the task tree. The sample contains a job with a script that is used to collect information about the requests and create an ASCII file with the information. A repository is also created during the import (SAMPLE_REQUEST_COMPLETE). The repository constant "FILENAME" contains the folder and file name of the file that is created. You can modify the repository constant if you want to change this. Also make sure that you include a reference from the identity store properties to the imported task.

To perform more complex actions, modify the script in the job.

2.5.34 Statement Execution Analysis

To help analyze the performance of SQL queries in the configuration, it is possible to log all statements that take longer than a predefined time to execute.

The threshold is defined with the package constant `MX_LOG_EXEC_THRESHOLD`. The value of this constant is the number of milliseconds to be used as a threshold. All queries taking longer than the specified value are logged.

The statement execution log can be accessed with the view `idmv_exec_stat`. The log is available on the [Statement Execution](#) tab in the Identity Management Administration User Interface, provided that the logged-in user has the privilege `MX_PRIV:WD:TAB_THRESHOLD`.

For more information, see the document *SAP Identity Management Solution Operations Guide*.

Related Information

[SAP Identity Management Solution Operation Guide](#)

2.5.35 Tracing

You can enable tracing to help troubleshoot specific situations. With tracing enabled, you can follow all operations performed on a specific entry.

Tracing is enabled for one entry at a time by setting the package constant `MX_TRACE_ENTRY` to the MSKEY or `<MSKEYVALUE>`. The trace is stored in the database table `mc_trace_data` with the view `idmv_trace_data`. The trace is available on the [Trace](#) tab in the Identity Management Administration User Interface, provided that the logged-in user has the privilege `MX_PRIV:WD:TAB_TRACE`.

If you want to include trace information from the Runtime Engine, the package constant `MX_TRACE_RT` must be set to `TRUE`.

For more information, see the document *SAP Identity Management Solution Operations Guide*.

Related Information

[SAP Identity Management Solution Operation Guide](#)

3 Configuring SAP Identity Management

This section provides the following information:

- How to use SAP Identity Management Developer Studio for developing the configuration for the identity management solution.
- How to use the Provisioning framework for SAP Identity Management 8.0.
- How to create repositories in the Identity Management Administration User Interface.
- How to maintain identity store entries, such as persons, roles, privileges and dynamic groups in the Identity Management User Interface.
- How to configure and implement self-service password reset.

Related Information

[SAP Identity Management Developer Studio \[page 108\]](#)

[Provisioning Framework for SAP Identity Management 8.0 \[page 399\]](#)

[Maintaining Repositories \[page 492\]](#)

[Maintaining Identity Store Entries \[page 501\]](#)

[Self-Service Password Reset \[page 553\]](#)

3.1 SAP Identity Management Developer Studio

SAP Identity Management Developer Studio is an Eclipse plug-in that provides the environment for developing the configuration for the identity management solution.

Identity Management Developer Studio offers the following features:

- Multi-user environment
- Authentication of users
- Configuration authorization
- Version control of configuration changes
- Graphical process editing
- Transport of configuration changes between Dev/QA/Prod system landscapes

The following sections describe how to maintain identity center properties, identity stores, identity store schema, packages, repository types, processes, forms, jobs and others.

Related Information

[Logging on to the Identity Management Developer Studio \[page 109\]](#)

[Switching Users in the Identity Management Developer Studio \[page 110\]](#)

[Authentication and Authorization \[page 111\]](#)

3.1.1 Logging on to the Identity Management Developer Studio

Prerequisites

- Eclipse version 2024-09 is required. We recommend you to choose the package *Eclipse IDE for Java Developers*.
- Identity Management Developer Studio plugin is installed.
- Connection to the Identity Management database is configured.
- The initial administrator user of the Identity Management Developer Studio exists in the database and in the User Management Engine (UME). In addition, it is assigned the `idm_authenticated_studio` UME action through a UME role. You can use the `idm.developer` role, as it already contains this action. For more information, see *Assigning idm_authenticated_studio UME action*.
- All the Identity Management Developer Studio users that are added by the initial administrator exist in UME and have the `idm_authenticated_studio` UME action assigned to them through a UME role.

Context

To log on to the Identity Management Developer Studio, proceed as follows:

Procedure

1. Start your Eclipse IDE.
2. In the main menu, choose **Window** > **Open Perspective** > **Other...** and select the *SAP Identity Management Developer Studio* perspective.
3. In the tree view, select the *Root* node.
4. In the tree view, expand the node for the Identity Management database.
5. Enter your credentials to log on.

→ Tip

If you open too many views in the *SAP Identity Management Developer Studio* perspective and your views get mixed up, you can reset the perspective to its original layout. To do so, choose **Window > Reset Perspective > Yes**.

📘 Note

The default timeout of your web session is 30 minutes or the global one of AS Java if configured. You can configure the session timeout of `idmdevstudio` Web module of AS Java to extend the expiration period and prevent relogin due to inactivity. For more information, see *Configuring Additional Settings*.

Related Information

[Installing the Identity Management Developer Studio](#)
[Initial Configuration of Identity Management Developer Studio](#)
[Assigning `idm\_authenticated\_studio` UME action](#)
[Configuring Additional Settings for SAP NetWeaver 7.3](#)
[Configuring Additional Settings for SAP NetWeaver 7.3 EHP1](#)
[Configuring Additional Settings for SAP NetWeaver 7.4](#)

3.1.2 Switching Users in the Identity Management Developer Studio

Switch User allows a user to log out of the Identity Management Developer Studio and switch to another user to log on.

Context

To switch user in the Identity Management Developer Studio, proceed as follows:

Procedure

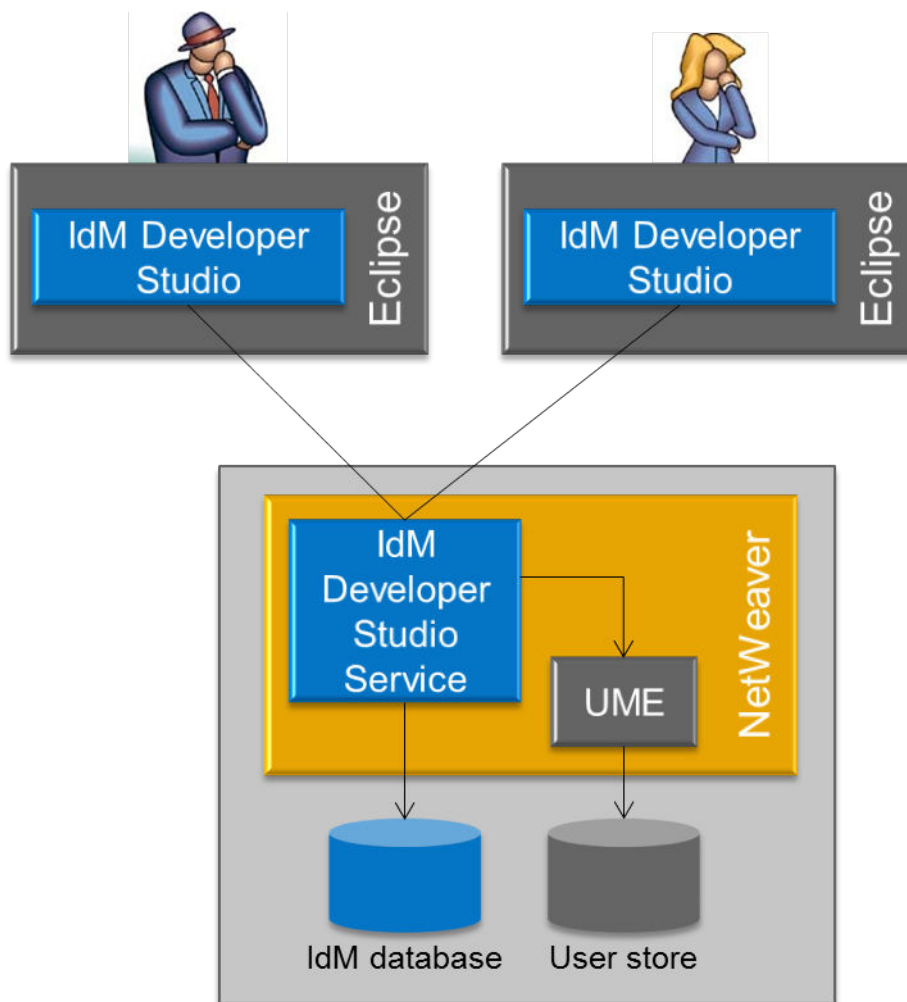
1. In the tree view, select the Identity Management database.
2. Choose *Switch User* from the context menu.
3. The Log on dialog box prompts for the credentials of the other user.

3.1.3 Authentication and Authorization

To work with the configuration in Identity Management Developer Studio, users need to be authenticated and granted the right authorizations.

Authentication

The authentication of users is handled by the User Management Engine (UME). Users must be defined in the UME in order to work with the configuration in the Identity Management Developer Studio. See the diagram below.



Authorization

To access a package, users must have the right authorizations. Below are listed the two types of Identity Management Developer Studio users with their key authorization differences:

- [Developer](#) - The users have access to one or more packages.
- [Developer Administrator](#):
 - Manage Identity Center's properties, housekeeping and users
 - Manage Dispatcher's properties, policy and jobs
 - Create Identity Store
 - Modify Identity Store schema
 - Manage package access rights

The following authorizations can be given to users:

- [View](#): The users can view the configuration, but cannot make any changes (only read and no write access).
- [Developer](#): The users have both read and write access to the configuration. Users with this authorization are able to change any value within the identity store, including the privilege assignments.
- [Layout Developer](#): The users are allowed to check out the package for editing, but only allowed to modify User Interface related configuration (e.g. the User Interface layout of tasks or display names).
- [Import](#): The users are allowed to import a new version of the package.
- [Owner](#): The users have owner privileges, such as the possibility to add, remove and modify the package authorizations in the Identity Management Developer Studio (on the [Authorized users](#) tab in the package properties)

Note

If the authorizations of a user have been changed, the user needs to log on again, by choosing "Switch User" from the context menu of the Identity Management database.

Related Information

[Switching Users in the Identity Management Developer Studio \[page 110\]](#)

[Adding an Identity Management Developer Studio User \[page 118\]](#)

[Authorized Users \(Packages\) \[page 1009\]](#)

[Creating an Additional Developer Administrator](#)

3.1.4 Maintaining Identity Center Properties

3.1.4.1 Viewing and Configuring Identity Center Properties

You can view and configure the Identity Center properties.

Context

To configure the Identity Center properties, do the following:

Procedure

1. Select the Identity Center node in the tree view and choose *Properties* from the context menu.
2. Select a tab to enter information about the identity store:
 - General
 - Housekeeping
 - Users

Related Information

[Configuring General Identity Center Properties \[page 114\]](#)

[Configuring Scheduled Procedures for Housekeeping \[page 116\]](#)

[Configuring Identity Management Developer Studio Users \[page 117\]](#)

3.1.4.2 Configuring General Identity Center Properties

Context

To configure the General Identity Center properties, proceed as follows:

Procedure

1. Select the Identity Center node in the tree view and choose  *Open*  from the context menu.
The *Identity Center* tab opens.
2. Select the *General* tab to enter information about the identity store.

Related Information

[Identity Center Properties \[page 965\]](#)

[Configuring the System Log Level for an Identity Center \[page 114\]](#)

[Configuring Job Timeouts for an Identity Center \[page 115\]](#)

[Configuring Import Options for an Identity Center \[page 116\]](#)

3.1.4.2.1 Configuring the System Log Level for an Identity Center

Context

To specify the log level for the system log, proceed as follows:

Procedure

1. Select the Identity Center node in the tree view and choose  *Open*  from the context menu.
2. Select the *General* tab.

3. Select a value for *Log level* and *Enable trace* in the *System log* section. You can choose from the following values:

- *Info*
- *Warning*
- *Error*
- *Fatal*

Only log messages with a log level equal to or higher than the selected log level are included in the system log.

4. Choose  *File*  *Save*  to save the properties.

Related Information

[General \(Identity Center\) \[page 965\]](#)

3.1.4.2.2 Configuring Job Timeouts for an Identity Center

Context

To configure the timeouts, number of restarts and concurrent runtime engines that are used when processing jobs, proceed as follows:

Procedure

1. Select the Identity Center node in the tree view and choose  *Open*  from the context menu.
2. Select the *General* tab.
3. Specify timeout values for regular jobs and bootstrap jobs.
4. Choose  *File*  *Save*  to save the properties.

Related Information

[General \(Identity Center\) \[page 965\]](#)

3.1.4.2.3 Configuring Import Options for an Identity Center

Context

To configure default values that are used when importing a package:

Procedure

1. Select the Identity Center node in the tree view and choose  *Open*  from the context menu.
2. Select the *General* tab.
3. Select a dispatcher as *Default dispatcher* that will be used when importing jobs.
4. Choose  *File*  *Save*  to save the properties.

Related Information

[General \(Identity Center\) \[page 965\]](#)

3.1.4.3 Configuring Scheduled Procedures for Housekeeping

There are a number of stored procedures that performs housekeeping actions like cleaning up logs and reconciling dirty entries. A procedure can have either one, two or no parameters, and some of them can be configured for the scheduled procedures listed here.

Prerequisites

At least one dispatcher should be configured to execute the *Housekeeping actions*. To ensure high availability, both dispatchers should be configured to execute the housekeeping.

Context

To configure the parameters and the scheduling of the procedures, proceed as follows:

Procedure

1. Select the Identity Center node in the tree view and choose  **Open**  from the context menu.
2. Select the *Housekeeping* tab.
All scheduled procedures are listed in the table.
3. You can perform the following operations:
 - a. Enable or disable the scheduled procedure.
 - b. Change the value of a parameter.
 - c. Change the schedule rule of a procedure.
 - d. Execute a scheduled procedure by selecting it in the list and choosing *Run now* from the context menu.

Related Information

[Housekeeping \(Identity Center\) \[page 968\]](#)

[Configuring the Dispatcher Policy \[page 390\]](#)

3.1.4.4 Configuring Identity Management Developer Studio Users

You can view and configure all users that are allowed to access Identity Management configuration Developer Studio.

Prerequisites

- To see the tab, you must be defined as a *Developer Administrator* user.

Context

To configure users who can log on to Identity Management Developer Studio:

Procedure

1. Select the Identity Center node in the tree view and choose  **Properties**  from the context menu.
2. Select the **Users** tab.
3. The list shows all the users defined in the Identity Center. To configure users, choose the following buttons on the right: **New**, **Modify** and **Deactivate**.

Related Information

[Users \(Identity Center\) \[page 971\]](#)

[Adding an Identity Management Developer Studio User \[page 118\]](#)

[Viewing Details About an Identity Management Developer Studio User \[page 120\]](#)

[Modifying Identity Management Developer Studio User \[page 121\]](#)

[Deactivating an Identity Management Developer Studio User \[page 122\]](#)

[Activating an Identity Management Developer Studio User \[page 123\]](#)

3.1.4.4.1 Adding an Identity Management Developer Studio User

You need to add users so that they can log on to Identity Management Developer Studio. The initial admin user was added when the database was installed or upgraded.

Prerequisites

- To see the tab, you must be defined as a **Developer Administrator** user.
- The user must be added to UME before it can be added to the list of Identity Center users. The user also needs the `idm_authenticated_studio` UME action assigned to it through a UME role. You can use the `idm.developer` role, as it already contains this action. For more information, see *Assigning idm_authenticated_studio UME action*.
- If an initial password is defined for the user that is added to UME, it must be changed so that the user can log on to the Identity Management Developer Studio.

Context

To add a user:

Procedure

1. Select the Identity Center node in the tree view and choose  *Properties* from the context menu.
2. Select the *Users* tab.
3. Choose *New*.
4. Fill in the fields in the *Add User* dialog box:
 - a. Enter the name of a user that exists in UME.

Note

Make sure that the name you enter is exactly the same as the user name in UME.

- b. Select a user type for the user:
 - *Developer*
A regular user that can be given authorization to packages.
 - *Developer Administrator*
An administrator who is allowed to create package, add users, add identity stores, modify identity store schemas (entry types/attributes).
5. Choose *OK* to close the dialog box.

The user name is verified against UME, and an error message is displayed if the names do not match.

Results

The user is added to the list and can be given access to packages.

Related Information

[Users \(Identity Center\) \[page 971\]](#)

[Providing Authorizations to a Package \[page 196\]](#)

[Assigning idm_authenticated_studio UME action](#)

3.1.4.4.2 Viewing Details About an Identity Management Developer Studio User

You can view details about an Identity Management Developer Studio user.

Prerequisites

- To see the tab, you must be defined as a *Developer Administrator* user.

Context

To view the details about a user, proceed as follows:

Procedure

1. Select the Identity Center node in the tree view and choose  *Properties*  from the context menu.
2. Select the *Users* tab.
The list shows all users defined in the Identity Center.
3. Select the user in the list and choose *Modify User...* from the context menu.
The *User Information* dialog box appears.
4. You see the details about the user, including which authorizations the user has and to which packages.
5. Choose *Cancel* to close the dialog box.

Related Information

[Users \(Identity Center\) \[page 971\]](#)

[Configuring Identity Management Developer Studio Users \[page 117\]](#)

[Modifying Identity Management Developer Studio User \[page 121\]](#)

3.1.4.4.3 Modifying Identity Management Developer Studio User

You can modify the user type for an Identity Management Developer Studio user.

Prerequisites

- To see the tab, you must be defined as a *Developer Administrator*.

Context

If a user changes roles and requires either more or fewer authorizations in the system, you can change the user's user type.

Procedure

1. Select the Identity Center node in the tree view and choose  *Properties* from the context menu.
2. Select the *Users* tab.
3. Select the user in the list and choose *Modify User...* from the context menu.
The *User Information* dialog box appears.
4. Change *User Type* for the user.
5. Choose *OK* to close the dialog box.

You have to confirm that you want to modify the user information.

Related Information

[Users \(Identity Center\) \[page 971\]](#)

[Adding an Identity Management Developer Studio User \[page 118\]](#)

[Configuring Identity Management Developer Studio Users \[page 117\]](#)

3.1.4.4.4 Deactivating an Identity Management Developer Studio User

Users that should not be able to log on any more with Identity Management Developer Studio can be deactivated.

Prerequisites

- To see the tab, you must be defined as a *Developer Administrator*.

Context

Note

When you deactivate a user, all authorization to packages are removed. These are not restored if the user is activated again.

To deactivate a user, proceed as follows:

Procedure

1. Select the Identity Center node in the tree view and choose  *Properties*  from the context menu.
2. Select the *Users* tab.
3. Select an active user in the list and choose *Deactivate User* from the context menu.
4. Confirm that you want to deactivate the user.

Related Information

[Users \(Identity Center\) \[page 971\]](#)

[Adding an Identity Management Developer Studio User \[page 118\]](#)

[Configuring Identity Management Developer Studio Users \[page 117\]](#)

[Activating an Identity Management Developer Studio User \[page 123\]](#)

3.1.4.4.5 Activating an Identity Management Developer Studio User

You can activate a user that has been deactivated.

Prerequisites

- To see the tab, you must be defined as a *Developer Administrator*.

Context

Note

Any authorizations revoked when the user was deactivated will not be restored.

To activate a user:

Procedure

1. Select the Identity Center node in the tree view and choose  *Properties*  from the context menu.
2. Select the *Users* tab.
3. Select a deactivated user in the list and choose *Activate User* from the context menu.
4. Confirm that you want to activate the user.

Related Information

[Users \(Identity Center\) \[page 971\]](#)

[Adding an Identity Management Developer Studio User \[page 118\]](#)

[Configuring Identity Management Developer Studio Users \[page 117\]](#)

[Deactivating an Identity Management Developer Studio User \[page 122\]](#)

3.1.5 Maintaining Identity Stores

3.1.5.1 Adding an Identity Store

Context

The Identity Center can contain one or more identity stores. The identity store is a central repository for managing identity information.

There can be several identity stores for different purposes, for example for different organizations or for staging, but only one identity store is connected to the Identity Management User Interface. If you need an Identity Management User Interface to another identity store, this requires a separate instance of SAP NetWeaver AS Java.

To create an identity store, proceed as follows:

Procedure

1. Select the *Identity Management database* node in the tree view and choose **► New ► Identity Store ►** from the context menu. In the *Create New Identity Store* dialog box, enter the name of the new identity store and provide the name of the Developer Administrator user.

The Developer Administrator user receives Developer Administrator rights for this identity store in the SAP Identity Management Developer Studio, and rights for the Identity Management User Interface and Identity Management Administration User Interface connected to this identity store.

The identity store is created with default values, and the corresponding node is added to the tree view.

2. Rename the identity store in the tree view and open the properties to modify the identity store properties.
3. The default properties of the identity store are shown in the properties view.
4. Select a sub-tab to enter information about the identity store:
 - *General*
 - *Password policy*

Related Information

[Identity Store \[page 74\]](#)

[Identity Store Properties \[page 971\]](#)

[General \(Identity Store\) \[page 972\]](#)

[Password Policy \(Identity Store\) \[page 973\]](#)

3.1.5.2 Importing an Identity Store

Context

Importing an identity store is relevant for an upgrade scenario from Identity Management 7.2 to 8.0, as only Identity Management 7.2 identity stores can be exported and then imported into the Identity Management 8.0 database using the Identity Management Developer Studio.

To import an Identity Management 7.2 identity store, proceed as follows:

Procedure

1. In the Identity Management Developer Studio tree view, select the *Identity Management database* node and choose *Import...* from the context menu.
2. Navigate to the .MCC file of the identity store you want to import and choose *Open*.
3. In the *Identity Store File (.mcc) Information* dialog box, enter a reason for the import (optional) and choose *OK*.

Results

The Identity Management 7.2 identity store is imported and the corresponding node is added to the tree view.

3.1.5.3 Modifying an Identity Store

Context

To modify an identity store, do the following:

Procedure

1. To modify the properties of the identity store, select the identity store in the tree view and choose  *Open*  from the context menu.

The *Identity store properties* tab is opened.

2. Select a sub-tab to enter information about the identity store:
 - *General*
 - *Password policy*

Related Information

[Identity Store \[page 74\]](#)

[Identity Store Properties \[page 971\]](#)

[General \(Identity Store\) \[page 972\]](#)

[Password Policy \(Identity Store\) \[page 973\]](#)

3.1.5.3.1 Configuring General Parameters for an Identity Store

Context

To configure the general parameters for an identity store, do the following:

Procedure

1. Select an identity store in the tree view and choose *Open* from the context menu.
2. Fill in the fields on the *General* tab.
3. Choose  *File*  to save the properties.

Related Information

[Modifying an Identity Store \[page 125\]](#)

[General \(Identity Store\) \[page 972\]](#)

3.1.5.3.2 Adding a User to the Identity Store

Context

When configuring an identity store, you can add a user to it manually. Normally this will be done with a job or by a task in the Identity Management User Interface, but adding the user manually, makes it easy, for example, to create an administrator user that is able to log on to the Identity Management User Interface. The user must also be added to UME with the same identifier and given the necessary actions as described in the document *Initial Configuration of Identity Management User Interface*.

Procedure

1. Select the identity store in the tree view and select the [General](#) tab in the properties view.
2. Choose [Add user...](#) .
The [Add identity store user](#) dialog box is displayed.
3. Fill in the fields with the following information:

Option	Description
User name	Enter a unique name for the user to be added.
Add manager privileges	Select this check box to give access to manager tabs in the Identity Management User Interface. For more information, see <i>Access to the Identity Management User Interfaces (URLs)</i> in <i>SAP Identity Management Security Guide</i> and <i>Adding User(s) to the Identity Store</i> in <i>Initial Configuration of Identity Management User Interface</i> .
Add administrator privileges	Select this check box to give access to administrator tabs in the Identity Management User Interface. For more information, see <i>Access to the Identity Management User Interfaces (URLs)</i> in <i>SAP Identity Management Security Guide</i> and <i>Adding User(s) to the Identity Store</i> in <i>Initial Configuration of Identity Management User Interface</i> .

4. Choose [OK](#) to close the dialog box.

Results

The user is added to the identity store and will be visible in the list of users in the Identity Management User Interface. For the users to be able to log on, they must be added to UME with a password.

Related Information

[General \(Identity Store\) \[page 972\]](#)

[Modifying an Identity Store \[page 125\]](#)

[Initial Configuration of Identity Management User Interface](#)

[Access to the Identity Management User Interfaces \(URLs\)](#)

[Adding User\(s\) to the Identity Store](#)

3.1.5.3.3 Configuring the Password Policy of an Identity Store

Context

For information about password provisioning, see section 8.3 in the *SAP Identity Management Security Guide*.

To configure the password policy for an identity store, proceed as follows:

Procedure

1. Select an identity store in the tree view and choose [Open](#) from the context menu.
2. Fill in the fields on the [Password policy](#) tab.
3. Choose  [File](#)  [Save](#)  to save the properties.

Next Steps

Use the properties of the MX_PASSWORD attribute to configure the validation of the password.

Related Information

[Password Policy \(Identity Store\) \[page 973\]](#)

[Modifying an Identity Store \[page 125\]](#)

[Adding or Modifying an Identity Store Attribute \[page 173\]](#)

3.1.5.3.4 Defining a Password Dictionary

Context

When you configure the password policy for an identity store, you can specify a password dictionary containing words that are illegal as passwords.

Procedure

1. Select the identity store in the tree view and select the *Password policy* tab in the property view.
2. Select *Use password dictionary*.
3. Define the password dictionary by entering words manually or import an existing dictionary.

Related Information

[Password Policy \(Identity Store\) \[page 973\]](#)

[Configuring the Password Policy of an Identity Store \[page 128\]](#)

3.1.5.3.4.1 Editing a Password Dictionary

Context

You can edit the password dictionary manually.

Procedure

1. Select the identity store in the tree view and select the *Password policy* tab in the property view.
2. Choose *Edit...* to the right of the *Use password dictionary*.
The *Password dictionary* dialog box is opened.
3. The current dictionary is displayed in the list.

- a. Add a word by typing the word in the *Word* field and choosing *Add*.
 - a. Modify a word by selecting the word in the list, modifying the word and choosing *Modify*.
 - a. Delete a word by selecting a word in the list and choosing *Delete*.
4. Choose *OK* to close the dialog box.

Related Information

[Password Policy \(Identity Store\) \[page 973\]](#)

[Configuring the Password Policy of an Identity Store \[page 128\]](#)

3.1.5.3.4.2 Importing a Password Dictionary

Context

You can import a file containing a word list to create a password dictionary.

Note

This must be a text file with one word per line.

Procedure

1. Select the identity store in the tree view and select the *Password policy* tab in the property view.
2. Choose *Import...* to the right of the *Use password dictionary*.
The *Import password dictionary* dialog box is opened.
3. Locate and select the file containing the word list.
4. Choose *Open*.
All words contained in the file are added to the dictionary

Next Steps

Choose *Edit...* to open the *Password dictionary* dialog box to verify that the word list contains the words from the imported file.

Related Information

[Password Policy \(Identity Store\) \[page 973\]](#)

[Configuring the Password Policy of an Identity Store \[page 128\]](#)

3.1.5.3.4.3 Exporting a Password Dictionary

Context

You can export the password dictionary to a text file containing one word per line.

Procedure

1. Select the identity store in the tree view and select the *Password policy* tab in the property view.
2. Choose *Export...* to the right of the *Use password dictionary*.
The *Export password dictionary* dialog box is displayed.
3. Locate the folder where you want to store the file.
4. Enter a file name.
5. Choose *Save* to save the password dictionary to a file.

Related Information

[Password Policy \(Identity Store\) \[page 973\]](#)

[Defining a Password Dictionary \[page 129\]](#)

3.1.5.4 Using Views to Access Identity Store Information

When accessing information in the identity store with an SQL query, it is recommended to use one of the default views. All views delivered with the product starts with idmv_.

Note

When using Microsoft SQL Server, it is recommended that all SELECT queries should include "with (nolock)". This will reduce the table locking, improve performance, and reduce the probability of deadlocks. For instance:

```
SELECT <columns> FROM <view> WITH (NOLOCK) WHERE <column>='<value>'
```

Naming syntax

The following naming syntax is used for the new views:

```
idmv_<dataset>_<subset>_<filter>
```

Dataset

This part specifies which main table is used for the view:

Name	Table	Contents
approvals	mxi_approval	Retrieves approvals.
approvers	mxi_approver	Retrieves approvers.
entry	mxi_entry	Retrieves entries.
exec_stat	mc_exec_stat	Retrieves information about statement execution.
link	mxi_link	Retrieves reference attributes.
linkaudit	mxi_link_audit	Retrieves audit information for reference attributes.
members	mxi_link + mxi_entry	Retrieves privilege/role members.
ovalue	mxi_old_values	Retrieves historical values from mxi_old_values.
owners	mxi_link + mxi_entry	Retrieves privilege/role owners.
repository_vars	mc_repository_vars	Retrieves repository variables.
trace	mc_trace_data	Retrieves trace information.
vallink	mxi_link + mxi_entry	Retrieves records from both mxi_values and mxi_link.

Note

As this is more costly, it is recommended to use either Value or Link in the case where you know what you are looking for.

Name	Table	Contents
value	mxi_values	Retrieves regular (non-reference) attributes from mxi_values.

Subset

This part indicates which subset of the table is used. The further down this list, the more costly the view. For performance reasons you should always use the least costly view that provides enough information.

Name	Comment
simple	Only the table.
basic	Includes some attribute names and/or entry types.
ext	Includes view specific information from joining with additional tables. Which tables is specified for each view.

Filters

This indicates if there is a filter on the view:

Name	Comment
<None>	All entries from the table are included, excluding entries which are marked as deleted and entries which are not currently valid (i.e. outside validity).
all	No filter, i.e. returns all entries.
active	Includes only the active entries.
inactive	Includes only the inactive entries.

View Overview

The following table contains views available in SAP Identity Management 8.0.

Each view is described in a separate topic. There may be columns that are not documented. These columns are deprecated and should normally not be used.

View	Comment
idmv_approvals_basic [page 135]	Returns all approvals by joining the tables mxi_approver and mxi_approval.
idmv_approvals_ext [page 135]	Same as above, but joins with mxi_entry to get MSKEYVALUES.
idmv_approvers_basic [page 137]	Returns all entries that have been involved in approvals. Filter on mcType and mcApprovalId to find approvers for a given approval.
idmv_approvers_ext [page 137]	Same as above, but joins with mxi_entry to get MSKEYVALUE and display name of the involved parties.

View	Comment
idmv_entry_simple [page 137]	Returns all active entries in the identity store from the table mxi_entry.
idmv_entry_simple_all [page 137]	Returns all entries from the table mxi_entry.
idmv_entry_simple_inactive [page 137]	Returns all inactive entries from the table mxi_entry.
idmv_exec_stat [page 139]	Provides information for statement execution analysis.
idmv_linkaudit_basic [page 139]	Returns audit for reference attributes from the table mxi_link_audit.
idmv_linkaudit_ext [page 139]	Same as above, but joins with mxi_entry to get MSKEY-VALUE and display name of the involved parties.
idmv_linkaudit_simple [page 139]	Same as idmv_link_audit_basic, but joins with the entry table to include information about the responsible for the operation.
idmv_link_basic_active [page 141]	Returns reference attributes and joins with mxi_attributes to get the attribute name.
idmv_link_ext [page 141]	Returns all reference attributes. Joins the tables mxi_attributes, mxi_objectclasses and mxi_entry to include the MSKEYVALUE and entry type name of both references.
idmv_link_ext_active [page 141]	Same as idmv_link_ext, but returns only active entries.
idmv_link_ext2 [page 141]	Same as idmv_link_ext, but includes information about MSKEYVALUE and entry type of the context (if present).
idmv_link_simple_active [page 141]	Returns all active reference attributes.
idmv_link_simple_all [page 141]	Returns all reference attributes (both active and inactive). No tables are joined.
idmv_link_simple_inactive [page 141]	Returns all inactive reference attributes. No tables are joined.
idmv_members [page 148]	An expanded list of all role/privilege members.
idmv_ovalue_basic_all [page 149]	All historic attribute values from mxi_old_values including attribute names.
idmv_ovalue_simple_all [page 149]	All values from mxi_old_values with no extra information.
idmv_owners [page 152]	An expanded list of all role/privilege owners.
idmv_repository_vars [page 153]	Returns all repository constants.
idmv_trace_basic [page 154]	Provides trace information.
idmv_trace_data [page 154]	Provides trace information. Joins with several other tables to retrieve
idmv_vallink_basic [page 156]	Returns all attributes. Joins with mxi_attributes to get attribute names. Excludes expired entries.
idmv_vallink_basic_active [page 156]	Joins with mxi_attributes. Excludes expired entries.
idmv_vallink_ext [page 156]	Returns all valid attributes from the mxi_values and mxi_link table. In addition it joins the tables mxi_attributes, mxi_changetype, mxp_audit, mxi_objectclasses.

View	Comment
idmv_vallink_ext_active [page 156]	Same as idmv_vallink_ext, but returns only active entries.
idmv_vallink_ext2 [page 156]	Same as idmv_vallink_ext, but also includes MSKEYVALUE of the entry and its reference attributes.
idmv_vallink_simple [page 156]	Returns all valid attributes from the mxi_values and mxi_link table.
idmv_value_basic [page 160]	Returns attribute values. Joins with mxi_attributes to get attribute names. Excludes expired entries.
idmv_value_basic_active [page 160]	Joins with mxi_attributes. Excludes expired entries.
idmv_value_basic_all [page 160]	Same as idmv_value_basic, but with no filter.
idmv_value_ext [page 160]	Joins with mxi_attributes, mix_objectclasses, mxi_audit and mxi_changetype. Excludes expired entries.
idmv_value_ext_active [page 160]	Excludes inactive entries.
idmv_value_ext_inactive [page 160]	Excludes active entries.
idmv_value_ext2 [page 160]	All current (non-expired) identity store entries.
idmv_value_simple_all [page 160]	No tables are joined.

3.1.5.4.1 Views for Approvals

This topic describes the views that return the approvals in the system:

- idmv_approvals_basic
- idmv_approvals_ext

The view idmv_approvals_ext is a superset of idmv_approvals_basic. The additional columns in idmv_approvals_ext are marked with "ext".

Column	View	Description
ApprovalID		Unique identifier for this approval.
ApprovalType		Indicates type of approval: Assignment approvals: 1: Requesting the assignment (Validate add task) 2: Requesting removal of assignment (Validate remove task) 3: Requesting validity change (Validate modify validity task) 200: Basic approval

Column	View	Description
ApproverMSKEY		MSKEY of the approver.
ApproverMSKEYVALUE	ext	The MSKEYVALUE of the approver.
AssignedByMSKEY		MSKEY of the user requesting the assignment.
AssignedByMSKEYVALUE	ext	The MSKEYVALUE of the user requesting the assignment.
ContextMSKEY		MSKEY of the context of the assignment.
ContextMSKEYVALUE	ext	The MSKEYVALUE of the context of the assignment.
IdS		The identity store this approval belongs to.
LinkID		Unique identifier of the assignment. (Only for assignment approvals.)
PVOMSKEY		MSKEY of entry to be approved. For assignment approvals, this will be the MSKEY of the pending value object (PVO).
TargetMSKEY		MSKEY of the role/privilege to be assigned.
TargetMSKEYVALUE	ext	The MSKEYVALUE of the role/privilege to be assigned.
TimeoutTime		The timeout for the approval. When the approval times out, it may be escalated, in which case a new TimeoutTime value is calculated.
UserMSKEY		MSKEY of the user getting the assignment. If PVOMSKEY is a pending value object, this is the user referenced by the PVO. Otherwise it will be the same as the PVOMSKEY. For non-assignment approvals, this will be the same as the PVOMSKEY.
UserMSKEYVALUE	ext	The MSKEYVALUE of the user getting the assignment.

3.1.5.4.2 Views for Approvers

This topic describes the views return all approvers in the system:

- idmv_approvers_basic
- idvm_approvers_ext

The view idmv_approvers_ext is a superset of idmv_approvers_basic. The additional columns in idmv_approvers_ext are marked with "ext".

Column	View	Description
ApprovalID		Unique identifier for this approval.
UserDISPLAYNAME	ext	The user's display name.
UserMSKEY		MSKEY of the user getting the assignment.
UserMSKEYVALUE	ext	The MSKEYVALUE of the user getting the assignment.
UserType		Indicates which role the given user has regarding this approval: 1: Assigner This is the user who requested the assignment. Only one per approval. Only for assignment approvals. 2: Target user This is the user who is receiving the assignment. Only one per approval. Only for assignment approvals. 3: Approver This is the approver(s) who have not yet approved or declined. Multiple users per approval. 4/5: Old approver This is the approver(s) who already has processed the approval. Multiple users per approval.

3.1.5.4.3 Views for Entries

This topic describes the columns in the views that return entries in the identity store:

- idmv_entry_simple
- idmv_entry_simple_all
- idmv_entry_simple_inactive

These views have the same columns. The only difference is the selection of entries contained in the views.

Column	Description
mcACEntry	Visibility of the entry: 0: Entry is visible to all 1: Entry is visible to owner and members 2: Entry is visible to owner only Same as the attribute MXAC_ENTRY
mcChangeCreated	This is the change number which created the entry. Will never be changed after the entry is created.
mcChangeNumber	Global counter containing the change number for the entry. This column is not updated when ulS_SetValue function is used.
mcCreated	The date the entry was created.
mcDirty	Indicates that the entry is dirty, and should be reconciled. 0: Entry OK >0: Dirty. Will be reconciled according to scheduled procedure
mcDirtyDate	Date of last update of mcDirty.
mcDisplayName	The entry's display name.
mcEntryOwner	Reference to repository which created this entry.
mcEntryState	Indicates the state of the entry. Will be one of the following values: 0: Active entry 1: Inactive entry (i.e. MX_INACTIVE is set) 2: Deleted entry 3: Entry pending deletion
mcEntryType	The entry's entry type.
mcEntryTypeld	Entry type ID.
mcIdStore	The ID of the identity store to which the entry belongs.

Column	Description
mcLastChangeOwner	Reference to repository which last updated this entry.
mcLastModified	The date when the entry was last updated. This column is not updated when ulS_SetValue function is used.
mcMSKEY	The entry's MSKEY, which is the unique identifier.
mcMSKEYVALUE	The entry's MSKEYVALUE.

3.1.5.4.4 Views for Statement Execution Analysis

This topic describes the view for statement execution analysis:

- idmv_exec_stat

Most columns in this view are optional and only filled in if the information is relevant.

Column	Description
ComponentId	The ID of the component logged the event. See Component-Name.
ComponentName	The name of the component that logged the event.
DateTime	When the event happened.
DisplayName	The display name of the entry being processed.
JobId	The Job ID of the job being executed.
JobName	The job's name.
mcDescription	Additional free text about the event.
mcExecTime	Number of milliseconds used for the operation per entry.
mcNumEntries	Number of entries.
mcStatement	Conditional or switch statement being executed.
MSKEY	The MSKEY of the entry being processed.
MSKEYVALUE	The MSKEYVALUE of the entry being processed.
TaskId	The Task ID of the task being executed.
TaskName	The task's name.

3.1.5.4.5 Views for Audit of Reference Attributes

This topic describes the columns in the following views that return the audit for reference attributes:

- idmv_linkaudit_basic
- idmv_linkaudit_ext
- idmv_linkaudit_simple

The views idmv_linkaudit_ext and idmv_linkaudit_simple are supersets of idmv_linkaudit_basic. The additional columns in idmv_linkaudit_ext and idmv_linkaudit_simple are marked with "ext" and "simple".

Column	View	Description
AdditionalInfo		Additional information about the audit.
AssignmentDisplayName	ext	The display name of the assigned role/privilege. (Only for assignments.)
AssignmentMSKEY		The MSKEY of the assigned role/privilege. (Only for assignments.)
AssignmentMSKEYVALUE	ext	The MSKEYVALUE of the assigned role/privilege. (Only for assignments.)
AuditDate		Date for the audit record.
AuditID		Reference to the audit table. -1 if no audit record for this event.
ContextDisplayName	ext	The context's display name. (Only for context based assignments.)
ContextMSKEY		The context's MSKEY. (Only for context based assignments.)
ContextMSKEYVALUE	ext	The context's MSKEYVALUE. (Only for context based assignments.)
DelegatedFrom	ext	Reference to the MSKEY of the user from which the automatic delegation of any approval is done. (Only used for automatic delegation.)
DelegatedTo		Reference to the MSKEY of the user receiving a delegated approval. Used for delegation and automatic delegation of approvals, as well as for adding and removing approvers.
LinkID		Unique identifier of the assignment.
NewValidFrom		Dates modified by approver. NULL means not changed.
NewValidTo		Dates modified by approver. NULL means not changed.
Operation		The numeric ID of the operation. See OperationText.
OperationText		The textual description of the operation.
Reason		Reason for requesting the assignment or reason added by approver.

Column	View	Description
ResponsibleDisplayName	ext simple	The display name of the user who performed the operation.
ResponsibleMSKEY		The MSKEY of the user who performed the operation.
ResponsibleMSKEYVALUE	ext simple	The MSKEYVALUE of the user who performed the operation.
UserDisplayName	ext	The display name of the user receiving the assignment/reference attribute.
UserMSKEY		The MSKEY of the user receiving the assignment/reference attribute.
UserMSKEYVALUE	ext	The MSKEYVALUE of the user receiving the assignment/reference attribute.

3.1.5.4.6 Views for Reference Attributes

This topic describes the columns in the views that return reference attributes:

- idmv_link_basic_active
- idmv_link_ext
- idmv_link_ext_active
- idmv_link_ext2
- idmv_link_simple_active
- idmv_link_simple_all
- idmv_link_simple_inactive

Some views contains extra columns, and this is indicated in the [View](#) column.

Column	View	Description
mcAddAudit		The AuditID of the Add member task. 0 or null: The task has not been executed -1: The task was not executed because it was not defined (on role/privilege or repository) -2: The task was not executed because this was a bootstrap job -3: The task was defined, but not executed because of a policy setting (add/del member policy) -4: The task was not executed, because a DIRECT_REFERENCE property was used in the assignment
mcAddedTime		When the reference was added.
mcAssignedDirect		Indicates if the reference was assigned directly. (0 or 1)
mcAssignedDynamicGroup		Indicates that the assignment was done through a dynamic group. Only valid for role assignments.
mcAssignedInheritCount		Number of parents from which this object was inherited. Used for role/privilege fan-in. 0 or a numeric value.
mcAssignedMasterPrivilege		Indicates that this assignment was a result of a privilege dependency policy.
mcAssigner		MSKEY of the user adding the reference.
mcAttrId		The attribute's unique identifier.
mcAttrName	basic_active ext ext_active ext2	The attribute's name.

Column	View	Description
mcAuditID		Reference to the audit ID which created this link. Only set for direct assignments. All assignments made in the same request, will have the same mcAuditID.
mcChangeNumber		Changenummer of last update. This will always be updated when mcModifyTime is updated.
mcContextAuto		Flag indicating that the context as a conditional context.
mcContextEntryType		Entry type of context entry.
mcCtxMSKEYVALUE	ext2	The MSKEYVALUE of the context.
mcCtxOcName	ext2	The name of the context entry type.
mcDelAudit		The AuditID of the Remove member task. See mcAddAudit.
mcDelRequestId		Same as mcRequestId, but used when deleting assignments.
mcDeprovAudit		The AuditID of the Deprovisioning task. See mcAddAudit.
mcDirty		Indicates whether the entry needs to be reconciled. 0 = Link is OK.
mcDirValidFrom		The validity of the direct assignment, which may be different from the inherited validity. The validity is the largest interval of the direct and the inherited.
mcDirValidTo		The validity of the direct assignment, which may be different from the inherited validity. The validity is the largest interval of the direct and the inherited.
mcDisabled		Indicates that the link belongs to an inactive entry.

Column	View	Description
mcExecState		Indicates the state of this assignment. 0: This is the default value, and will be set in the case where there are no add member event processes defined. 1: Add assignment OK. 2: Add assignment rejected. 4: Add assignment failed. 512: Add assignment still in process or pending. 1024: This is the default value, and will be set in the case where there are no remove member event processes defined. 1025: Remove assignment OK. 1026: Remove assignment rejected. 1028: Remove assignment failed. 1536: Remove assignment still in process or pending.
mcExecStateHierarchy		Indicates the accumulated state of this hierarchy assignment. Only for root entries. Can be a combination of the values 1, 2 and 4, as well as either 512 or 1024.
mcGroupGUID		GUID used for grouping of links.
mcLastAudit		The last audit ID, which is needed during setting of inactive and pending delete.

Column	View	Description
mcLinkState		Indicates the state of the entry. The following values are allowed: 0: Active link. Indicates that the assignment is active, and should be visible to any application viewing the assignments. 1: Inactive link. An inactive link may for example be a privilege which is assigned, but the AddMember task has not yet completed. 2: Deleted link. The link has been deleted. simple_active = 0 smple_inactive = 1 ext_active = 0
mcLinkID		Unique identifier for this reference.
mcLinkType		The type of reference 0: Unspecified entry reference (i.e. manager, secretary) 1: Hierarchy (i.e. role-role or privilege-role). 2: Assignment (i.e. user-role, user-privilege, user-group) 3: Owner (i.e. the MX_OWNER attribute)
mcModify		Timestamp when entry was modified.
mcModifyAudit		The AuditID of the Modify task. See mcAddAudit.
mcModifyNewValidFrom		New valid from. Used when a new validity is requested, before it is actually set.
mcModifyNewValidTo		New valid to. Used when a new validity is requested, before it is actually set.
mcModifyReason		User reason for modification.
mcModifyTime		When the reference was last modified.
mcModifyValidity		Timestamp when validity was updated.

Column	View	Description
mcNotAllowedFor		Indicates that there is an allowed-for clause on the object, and the user is not in the list.
mcNotifyAudit		The AuditID of the Notification task. See mcAddAudit.
mcOrphan		Indicates that the entry is an orphan one. This can happen, for example, if this assignment is inherited, and the parent is removed. Then the Remove member task for this entry will fail. For more information about orphan assignments, and how to revoke them using a DB stored procedure, see SAP Note 2499697 .
mcOtherEntryType		Entry type of mcOtherMSKEY.
mcOtherMSKEY		The referenced entry (for example, role), privilege or user.
mcOtherMSKEYVALUE	ext ext_active ext2	The MSKEYVALUE of the referenced entry.
mcOtherOcName	ext ext_active ext2	The name of the entry type of the referenced entry.

Column	View	Description
mcProcessInfo		This column is used by several processes when updating a value. Runtime engine: It can be set directly from a To identity store pass. In this case the default value is JobId=%\$ddm.mcjob%, but any information can be added here. Identity Management User Interface: Set to the MSKEY of the user which is logged in. Identity Management Developer Studio (Management Console as of version 7.2): Set to the text "GUI". Procedures (dynamic group expansion): Set to the text "DG". Procedures (reconciliation process): Set to the text "Reconcile".
mcProvAudit		AuditID of the Provisioning task. See mcAddAudit.
mcReason		Reason for adding the reference.
mcRepository		References the repository from which the value was written and is used to specify attribute ownership. Note This is not the repository of the assignment, but the repository from which the assignment was written. By default, there is no repository defined when writing from the Identity Management User Interface, but this can be set using the "User Interface Repository" field on the "General" tab of the identity store properties.
mcRequestId		The request identifier.
mcSoDViolation		A counter indicating the number of SoD (Separation of Duties) violations for this reference.

Column	View	Description
mcThisEntryType		Entry type of mcThisMSKEY.
mcThisMSKEY		The entry where the reference is defined.
mcThisMSKEYVALUE	ext ext_active ext2	The MSKEYVALUE of the entry containing the reference.
mcThisOcName	ext ext_active	The name of the entry type for the entry containing the reference.
mcUnsupportedContextType		Indicates that the link is denied, because there is a context on the link, but this context does not match any of the context types on the privilege.
mcValidateAddAudit		AuditID of the Validate add task. See mcAddAudit.
mcValidateDelAudit		AuditID of the Validate remove task. See mcAddAudit.
mcValidateModValAudit		AuditID of the Validate modify validity task. See mcAddAudit.
mcValidFrom		Validity of calculated assignment. NULL in either of these values indicate no validity.
mcValidTo		Validity of calculated assignment. NULL in either of these values indicate no validity.

3.1.5.4.7 Views for Role/Privilege Members

This topic describes the columns in the view returning the role/privilege members:

- idmv_members

If a member is a role or a privilege, this will also be expanded, so the view returns all the actual members.

Column	Description
EntryMskey	MSKEY of the role/privilege.

Column	Description
MemberMskey	MSKEYs of the each member of the role/privilege.

3.1.5.4.8 Views for Historic Values

This topic describes the columns in the views that return historic values in the identity store:

- idmv_ovalue_simple_all
- idmv_ovalue_basic_all

The view idmv_ovalue_basic_all is a superset of idmv_ovalue_simple_all. The additional columns in idmv_ovalue_basic_all are marked with "basic".

Column	View	Description
aLong		Value of the attribute if it is longer than 2000 characters.
Attr_ID		Unique identifier of the attribute.
AttrName	basic	The name of the attribute.
AuditId		AuditID which last updated the attribute.
aValue		The attribute value. This is the actual value, unless the value is greater than 2000 characters, in which case it is stored in aLong.
bChecksum		Numeric checksum of value, which is used to detect changes.
ChangedBy		The MSKEY of the user who last changed the attribute.
ChangeName	basic	The descriptive name of the change-type.
Changenum		Database internal unique change number.

Column	View	Description
Changetype		Type of change: 0: Modify 1: Delete 2: Insert 3: Disable 4: Enable
context		MSKEY of the context for a removed assignment.
CreateTime		This is the time when the attribute was last modified.
datatypeid	basic	The attribute's data type.
display_name	basic	The attribute's display name.
ExpiryTime		The time the attribute was removed.
IdAudit		Reference to the event process that is started as a result of changing this attribute. Only used for MX_PRIVILEGE and MX_ROLE.
IS_ID	basic	The ID of the identity store the attribute belongs to.
LinkId		LinkId of a deleted reference.
mcAssignedDirect		For assignments: This was assigned directly (0/1)
mcAssignedDynamicGroup		For assignments: Assigned as a result of a dynamic group (0/1)
mcAssignedInheritCount		For assignments: Inherit count (0..n)
mcReason		Reason for adding or modifying a reference attribute.
mcValidFrom		The specified Valid from date for a reference attribute.
mcValidTo		The specified Valid to date for a reference attribute. When an attribute is deleted, this value is written to Expiry-Time.

Column	View	Description
Modifytime		The time the attribute was last modified.
MSKEY		Unique identifier of the entry owning the attribute.
MultiValue	basic	Indicates that the attribute is multi-value. 1= Multivalue 0= Single value
OLD_ID		Unique identifier in the mxi_old_values table.
Owner_ID	basic	Indicates which repository owns this attribute.
ParentAuditId		AuditID of parent which last updated the attribute. If there is no parent, this is set to NULL.
ReferenceObjectClass	basic	If datatypeid is entry reference, this is the entry type which is referenced, or NULL if any entry type can be referenced.
SearchValue		Searchable value, i.e. upper case value of aValue.

Column	View	Description
UserID		This column is used by several processes when updating a value. Runtime engine: It can be set directly from a To identity store pass. In this case the default value is JobId=%\$ddm.mcjob%, but any information can be added here. Identity Management User Interface: Set to the MSKEY of the user which is logged in. Identity Management Developer Studio (Management Console as of version 7.2): Set to the text "GUI". Procedures (dynamic group expansion): Set to the text "DG". Procedures (reconciliation process): Set to the text "Reconcile".
ValOwner		References the repository from which the value was written and is used to specify attribute ownership. Note This is not the repository of the assignment, but the repository from which the assignment was written. By default, there is no repository defined when writing from the Identity Management User Interface, but this can be set using the "User Interface Repository" field on the "General" tab of the identity store properties.

3.1.5.4.9 Views for Role/Privilege Owners

This topic describes the columns in the view returning the role/privilege members:

- idmv_owners

Column	Description
EntryMskey	MSKEY of the role/privilege.
EntryMSKEYVALUE	MSKEYVALUE of the role/privilege.
OwnerMskey	MSKEYs of each owner of the role/privilege. If the owner for instance is a role, this will be the MSKEY of the role, not each role member.
OwnerExpandedMskey	The expanded list of MSKEYs of owners of the role/privilege. If the owner for instance is a role, this will be the MSKEYs of each member of the role.

3.1.5.4.10 Views for Repository Constants

This topic describes the columns in the view returning the repository constants:

- idmv_repository_vars

Column	Description
RepDisabled	Indicates that the repository is disabled.
RepName	The name of the repository.
VarCategory	One of the following values: -1: Repository constant -2: Repository variable (for future use)
VarName	The name of the repository constant.

Column	Description
VarType	The type of constant: 0: String 1: JDBC URL 2: LDAP URL 3: File 4: Password 5: Task reference 6: Boolean 7: Folder 8: JDBC Driver 9: OLE DB driver 10: Entry type 11: Attribute name 12: Repository name 13: Job reference 14: Entry reference
VarValue	The value of the repository constant.

3.1.5.4.11 Views for Trace Data

This topic describes the columns in the views that contain trace data:

- idmv_trace_basic
- idmv_trace_data

The view idmv_trace_data is a superset of idmv_trace_basic. The additional columns in idmv_trace_data are marked with "data".

Column	View	Description
AttributeID		The attribute's unique identifier.
AttributeName	data	The attribute's name.
AttributeValue		The attribute value.
AuditID		The operation's audit ID.

Column	View	Description
Changetype		Which type of change was being done 0: Add/Modify 1: Delete case sensitive 2: Delete case insensitive
ChildMSKEY		This is used in the case this is a pending value object being traced. The EntryMSKEY will then always be the MSKEY being traced.
Component	data	The name of the component.
ComponentID		The ID of the component logged the event. See Component for a description of the value.
DisplayName	data	The display name of the entry being traced.
EntryMSKEY		The MSKEY of the entry being traced.
Event	data	The descriptive name of the event being logged.
EventID		The ID of the event being logged. See Event for a description of the value.
ID		This is the unique identifier for this trace event.
IDOriginal		The original ID in case of a rollback. In this case, the runtime engine (or other component) will roll back a transaction, but re-insert the items afterwards. This value is then set to the original ID which was written.
JobID		The Job ID of the job involved in the trace event.
JobName	data	The name of the job.
Message		Additional text.
RepositoryID		The ID of the repository involved in the trace event (if any).
RepositoryName	data	The name of the repository.

Column	View	Description
Request		The request ID of the operation (if any).
TaskID		The ID of the task involved in the trace event (if any).
TaskName	data	The name of the task.
TaskSQL	data	SQL statement or procedure used in conditional tasks.
TaskType	data	The task type, for instance conditional or switch task.
TraceTime		When the trace event was written.

3.1.5.4.12 Views for Attribute Values

This topic describes the columns in the view returning attribute values:

- idmv_value_basic
- idmv_value_basic_active
- idmv_value_basic_all
- idmv_value_ext
- idmv_value_ext_active
- idmv_value_ext_inactive
- idmv_value_ext2
- idmv_value_simple_all

Some views contains extra columns, and this is indicated in the "View" column.

Column	View	Description
aLong	ext ext_active ext_inactive ext2 simple_all	Value of the attribute if it is longer than 2000 characters.
Attr_ID		Unique identifier of the attribute.
AttrName	not simple_all	The name of the attribute.

Column	View	Description
aValue		The attribute value. This is the actual value, unless the value is greater than 2000 characters, in which case it is stored in aLong.
bChecksum		Numeric checksum of value, which is used to detect changes.
ChangedBy		The MSKEY of the user who last changed the attribute.
ChangedByAudit		AuditID which last updated the attribute.
Changename	ext	Type of change:
	ext_active	0: Modify
	ext_inactive	1: Delete
	ext2	2: Insert
		3: Disable
		4: Enable
Changenumber		Database internal unique change number.
Changetype		The numeric value of the change type: 0: Modify 1: Delete 2: Insert 3: Disable 4: Enable
datatypeid	not simple_all	The attribute's data type.
Disabled	ext	Indicates whether the attribute is disabled or not.
	ext2	
	simple_all	
display_name	not simple_all	The attribute's display name.

Column	View	Description
ExecStatus		Indicates the accumulated task execution state of this hierarchy. Only for root entries. Can be a combination of the values 1, 2 and 4, as well as either 512 or 1024.
ExpiryTime		Time when attribute expires.
ExtAttribute	ext2	The attribute's display name.
ExtValue	ext2	The attribute's value. If the attribute is an entry reference, this is the MSKEY of the referenced entry. If the attribute is an audit flag reference, this is the audit flag. If the attribute is a task reference, this is the Task ID.
ExtMSKEY	ext2	MSKEYVALUE (MSKEY) of the entry the attribute belongs to.
is_id		The ID of the identity store the attribute belongs to.
Modifytime		The time the attribute was last modified.
MSKEY		Unique identifier of the entry owning the attribute.
MultiValue	not simple_all	Indicates that the attribute is multivalued. 1= Multivalued 0= Single value
ocld	ext ext_active ext_inactive ext2	Entry type ID.

Column	View	Description
ocName	ext	Entry type name.
	ext_active	
	ext_inactive	
	ext2	
Owner_ID	not simple_all	Indicates which repository owns this attribute.
ParentMSKEY		Contains the reference to the entry which "owns" the pending value object (PVO).
ProvStatus	ext	Provisioning status.
	ext_active	
	ext_inactive	
	ext2	
ReferenceObjectClass	not simple_all	If data type of the attribute (datatypeid) is entry reference, this is the entry type which is referenced, or NULL if any entry type can be referenced.
SearchValue		Searchable value, i.e. upper case value of aValue.
UserID		This column is used by several processes when updating a value. Runtime engine: It can be set directly from a To identity store pass. In this case the default value is JobId=%\$ddm.mcjob%, but any information can be added here. Identity Management User Interface: Set to the MSKEY of the user which is logged in. Identity Management Developer Studio (Management Console as of version 7.2): Set to the text "GUI". Procedures (dynamic group expansion): Set to the text "DG". Procedures (reconciliation process): Set to the text "Reconcile".

Column	View	Description
ValOwner		References the repository from which the value was written and is used to specify attribute ownership.

Note

This is not the repository of the assignment, but the repository from which the assignment was written. By default, there is no repository defined when writing from the Identity Management User Interface, but this can be set using the "User Interface Repository" field on the "General" tab of the identity store properties.

3.1.5.4.13 Views for All Attributes

This topic describes the columns in the view returning all attributes in the identity store:

- idmv_vallink_basic
- idmv_vallink_basic_active
- idmv_vallink_ext
- idmv_vallink_ext_active
- idmv_vallink_ext2
- idmv_vallink_simple

Some views contain extra columns, and this is indicated in the "View" column.

Column	View	Description
ExtAttribute	ext2	The attribute's display name.
ExtMSKEY	ext2	MSKEYVALUE (MSKEY) of the entry the attribute belongs to.

Column	View	Description
ExtValue	ext2	The attribute's value. If the attribute is an entry reference, this is the MSKEY of the referenced entry. If the attribute is an audit flag reference, this is the audit flag. If the attribute is a task reference, this is the Task ID.
mcAssignedDirect	ext ext_active ext2	Indicates the type of reference was assigned directly. (0 or 1.)
mcAssignedDynamicGroup	ext ext_active ext2	Indicates that the assignment was done through a dynamic group. Only valid for role assignments.
mcAttrAccessID	ext ext_active ext2	Indicates access to this attribute 1: Read write compare 2: Compare only
mcAttrDisplayName	not simple	The attribute's display name.
mcAttrId		Unique identifier of the attribute.
mcAttrName	not simple	The name of the attribute.
mcAttrOwnerID	ext ext_active ext2	Indicates which repository owns this attribute.
mcAuditID	ext ext_active ext2	Reference to the AuditID which created this reference.
mcChangedBy	ext ext_active ext2	The MSKEY of the user who last changed the attribute.

Column	View	Description
mcChangeName	ext	Type of change:
	ext_active	0: Modify
	ext2	1: Delete
		2: Insert
		3: Disable
		4: Enable
mcChangeNumber	ext	Database internal unique change number.
	ext_active	
	ext2	
mcChecksum	ext	Numeric checksum of value, which is used to detect changes.
	ext_active	
	ext2	
mcDataTyped	not simple	The attribute's data type.
mcExecState	ext	Indicates the accumulated task execution state of this hierarchy. Only for root entries. Can be a combination of the values 1, 2 and 4, as well as either 512 or 1024.
	ext_active	
	ext2	
mcExpiryTime	ext	Time when attribute expires.
	ext_active	
	ext2	
mcIdStore	not simple	The ID of the identity store the attribute belongs to.
mcLinkDISPLAYNAME	basic	Display name of the referenced entry.
	basic_active	
mcLinkMSKEY	basic	MSKEY of the referenced entry.
	basic_active	
mcLinkMSKEYVALUE	basic	MSKEYVALUE of the referenced entry.
	basic_active	

Column	View	Description
mcMasterMSKEY	ext	AuditID of parent which last updated the attribute. If there is no parent, this is set to NULL. About context variables.
	ext_active	
	ext2	
mcModifyTime	not simple	The time the attribute was last modified.
mcMSKEY		Unique identifier of the entry owning the attribute.
mcMultiValue	ext	Indicates that the attribute is multi-value. 1= Multivalue 0= Single value
	ext_active	
	ext2	
mcOtherOclId	ext	The entry type ID of the referenced entry.
	ext_active	
	ext2	
mcOtherOcName	ext	The name of the entry type of the referenced entry.
	ext_active	
	ext2	
mcParentMSKEY	ext	Contains the reference to the entry which "owns" the pending value object (PVO).
	ext_active	
	ext2	

Column	View	Description
mcProcessInfo	ext	This column is used by several processes when updating a value.
	ext_active	
	ext2	Runtime engine: It can be set directly from a To identity store pass. In this case the default value is Jobld=%\$ddm.mcjob%, but any information can be added here.
		Identity Management User Interface: Set to the MSKEY of the user which is logged in.
mcProvStatus	ext	Provisioning status.
	ext_active	
	ext2	
mcReference	basic	Indicates whether this is a reference attribute or not:
	basic_active	0: Regular attribute
		1: Reference attribute
mcRepository	ext	References the repository from which the value was written and is used to specify attribute ownership.
	ext_active	
	ext2	

Note

This is not the repository of the assignment, but the repository from which the assignment was written. By default, there is no repository defined when writing from the Identity Management User Interface, but this can be set using the "User Interface Repository" field on the "General" tab of the identity store properties.

Column	View	Description
mcSearchValue		Searchable value, i.e. upper case value of aValue.
mcValue		The attribute value, unless the value is greater than 2000 characters.
MSKEY	not simple	Unique identifier of the entry owning the attribute.

3.1.6 Maintaining the Identity Store Schema

3.1.6.1 Standard Entry Types

The identity store stores identity data according to a schema consisting of entry types and attributes. The entry types are objects that describe how the different identity-relevant objects are represented in the Identity Center. The entry types used when working with the SAP provisioning framework are:

- **MX_PERSON**
This is the entry type used for user objects in the system.
- **MX_ROLE**
This is the entry type used for business role objects. Nesting MX_ROLE entries is possible.
- **MX_PRIVILEGE**
This is the entry type used for permission objects (technical roles) in the system, including:
 - ABAP roles and profiles
 - Portal and UME roles
 - UME database groups
 - LDAP groups
 - SAP HANA™ database roles
 Nesting is not possible.
- **MX_GROUP**
This is the entry type used for LDAP and AS Java group hierarchies that contain privileges. In addition to being a privilege itself, an LDAP or AS Java group can for example contain privileges that represent ABAP roles, ABAP profiles, or portal roles. The MX_GROUP attribute contains the hierarchical structure used for assigning these privileges to the users.
For more information on the relationship between MX_GROUP and MX_PRIVILEGE, see *Group Concepts*.
- **MX_COMPANY_ADDRESS**
This is the entry type used for company addresses.
- **MX_HCM_EMPLOYEE**
This is an entry type used for employees read from the SAP HCM system. It allows for event-driven provisioning for SAP HCM identity data. It is included with the HCM staging area identity store.

These entry types are delivered with predefined sets of attributes that you can extend to meet your requirements.

Related Information

[Group Concepts \[page 72\]](#)

[Identity Store Schema](#)

3.1.6.2 Adding or Modifying an Entry Type

In addition to the entry types created automatically in the identity store, you can create your own entry types manually. You can also modify existing entry types.

Context

To add or modify an entry type in the identity store, do the following:

Procedure

1. Select an identity store's *Entry types* entry in the tree.
2. Choose the function:
 - To add an entry type, select *Entry types* in the tree and choose  *New*  *Entry type*  from the context menu.
 - To modify an entry type, select the entry type and choose *Properties* from the context menu.
3. The *Entry type* tab is opened.
4. Select a sub-tab to enter information about the attribute:
 - *General*
 - *Event tasks*
 - *Relations*
 - *Attributes*
5. Choose  *File*  *Save*  to save the attribute properties.

Related Information

[Entry Type Properties \[page 981\]](#)

[General \(Entry Type\) \[page 981\]](#)

[Event Processes \(Entry Type\) \[page 983\]](#)

[Relations \(Entry Type\) \[page 984\]](#)

[Attributes \(Entry Type\) \[page 985\]](#)

3.1.6.2.1 Configuring General Parameters for an Entry Type

Context

To configure the general parameters for an entry type, do the following:

Procedure

1. Open an identity store's *Entry types* node in the tree view.
2. Select an entry type and choose *Properties* from the context menu.
3. Fill in the fields on the *General* tab.
4. Choose ► *File* ► *Save* ► to save the properties.

Related Information

[General \(Entry Type\) \[page 981\]](#)

[Entry Types \[page 71\]](#)

3.1.6.2.2 Defining a Context Entry Type

Context

If you are using context based assignments, you define an entry type as a context entry type:

Procedure

1. Open an identity store's *Entry types* node in the tree view.
2. Select an entry type and choose *Properties* from the context menu.
3. Select the *General* tab.
4. Select the *Context entry type* check box to specify that this entry type will be used for context based assignments.

5. Choose  [File](#)  to save the properties.

Related Information

[General \(Entry Type\) \[page 981\]](#)

[Context Based Assignments \[page 60\]](#)

[Entry Types \[page 71\]](#)

3.1.6.2.3 Configuring Access Limitations for an Entry Type

Context

You can define access limitation for an entry type in the following way:

Procedure

1. Open an identity store's [Entry types](#) node in the tree view.
2. Select an entry type and choose [Properties](#) from the context menu.
3. Select the [General](#) tab.
4. Fill in the fields in the [Access limitations](#) group box.
5. Choose  [File](#)  to save the properties.

Note

Using access limitations will impact performance. For example when searching for roles and displaying role memberships, the access limitations must be checked for every role before displaying it. For more information, see *UI Configurations* section in *Factors that Influence Performance*.

Related Information

[General \(Entry Type\) \[page 981\]](#)

[Access Control \[page 33\]](#)

[Entry Types \[page 71\]](#)

[Factors that Influence Performance \[page 15\]](#)

3.1.6.2.4 Specifying Event Processes for an Entry Type

Context

When configuring an entry type, you use the [Event Processes](#) tab to configure the event processes of the entry type. Event processes can be used to automate provisioning of identity store entries. For example, event processes can create users in other repositories when new entries are added to the identity store. They can also perform specific actions if the location of an employee is modified for example, such as revoking an authorization and providing access to new systems.

Note

Adding event processes for an entry type can affect performance, and should only be used if strictly necessary.

Procedure

1. Open an identity store's [Entry types](#) node in the tree view.
2. Select an entry type and choose [Open](#) from the context menu.
3. Select the [Event Processes](#) tab.
4. On the [Event Processes](#) tab, define add, modify and/or delete event processes for the entry type. Use the ... button to the right of the field to select and add the event process. You can do the following:
 - For the [Add](#) a process operation, you can select a process of type [Add Entry Process](#).
 - For the [Modify](#) a process operation, you can select a process of type [Modify Entry Process](#).
 - For the [Delete](#) a process operation, you can select a process of type [Delete Entry Process](#).
5. Choose  [File](#)  [Save](#)  to save the properties.

Related Information

[Event Processes \(Entry Type\) \[page 983\]](#)

[Entry Types \[page 71\]](#)

3.1.6.2.5 Defining Relations for an Entry Type

Context

When you configure an entry type, you use the [Relations](#) tab to configure the relations for entries of this entry type.

By choosing [Allow subordinates](#), the MXREF_<This_entry_type> attribute is created in the identity store, which is used to reference this entry type from an entry, for instance when assigning a role with a custom entry type to a person.

For each entry type defined as a legal subordinate, the attribute MXREF_<This_Entry_Type> is defined as a legal attribute on the subordinate entry type. In addition, an MXMEMBER_<Subordinate_Entry_Type> attribute is created (unless it already exists), and added as a legal attribute on the entry type being created.

If selecting [Allow multiple superiors of this entry type](#), the created MXREF_<This_Entry_Type> will be defined as multi-value attribute, otherwise it is defined as single value.

For more information about the attributes, see the document *Identity Store Schema*.

Note

These settings cannot be changed after the entry type has been created

Procedure

1. Open an identity store's [Entry types](#) node in the tree view.
2. Select an entry type and choose [Properties](#) from the context menu.
3. Select the [Relations](#) tab.
4. Specify the rules for relations for the entry type.
5. Choose  [File](#)  [Save](#)  to save the properties.

Related Information

[Relations \(Entry Type\) \[page 984\]](#)

[Entry Types \[page 71\]](#)

[Identity Store Schema](#)

3.1.6.2.6 Defining Attributes for an Entry Type

Context

When you configure an entry type, you use the [Attributes](#) tab to define which attributes are allowed and which are mandatory for the entry type.

If you have selected [Automatically create attributes](#) for the identity store, attributes that are added to the identity store are added to the specified entry type as well.

Note

Never remove any of the attributes starting with MX_ from the system entry types starting with MX_.

Procedure

1. Open an identity store's [Entry types](#) node in the tree view.
2. Select an entry type and choose [Properties](#) from the context menu.
3. Select the [Attributes](#) tab.
4. Define the attributes for the entry type. The list contains all defined entry types. For each entry type you can specify whether or not the attribute is allowed or if it is mandatory. Specify the following parameters:
 - [Allow](#)
 - [Mandatory](#)
 - [List](#)
 - [Ext. properties](#)
5. Choose  [File](#)  [Save](#)  to save the properties.

Related Information

[Configuring Extended Attribute Properties \[page 172\]](#)

[Attributes \(Entry Type\) \[page 985\]](#)

[Identity Store \[page 74\]](#)

3.1.6.2.6.1 Configuring Extended Attribute Properties

Context

When you configure attributes for an entry type or entry types for an attribute, you use the [Extended attribute properties](#) dialog box to configure properties for multivalue reference attributes, like roles or privileges. You decide which information should be available in the Identity Management User Interface when viewing or modifying an assignment.

Note

If you have used the Assignment request task to create the assignment, make sure that the parameters of that task match this configuration.

Procedure

1. View the properties of an entry type and select the [Attributes](#) tab or view the properties of an attribute and select the [Entry types](#) tab.
2. Select the ... button behind one of the multivalue reference attributes, for instance MXREF_MX_ROLE to open the [Extended attribute properties](#) dialog box.
3. Fill in the fields in the dialog box.
4. Choose **OK** to close the dialog box and to save the properties.

Related Information

[Defining Entry Types for an Identity Store Attribute \[page 185\]](#)

[Entry Types \(Attribute\) \[page 1001\]](#)

[Extended Attribute Properties \[page 985\]](#)

[Attributes \(Entry Type\) \[page 985\]](#)

[Defining Attributes for an Entry Type \[page 171\]](#)

3.1.6.3 Standard Schema Attributes

The schema used by the SAP provisioning framework contains a number of attributes that are used to describe the entry types (for example, MX_LASTNAME, MX_FIRSTNAME). See the identity store schema for a complete list of the attributes available.

Some of the most important are shown in the table below.

Attributes

Attribute	Description	Applicable Entry Type
MSKEYVALUE	Unique identifier for the identity object.	All
ACCOUNT<Repository>	Unique user ID for the user in the target repository.	MX_PERSON
MX_REPOSITORYNAME	Identifier for the home repository where the original privilege is defined.	MX_PRIVILEGE

Related Information

[Identity Store Schema](#)

3.1.6.4 Adding or Modifying an Identity Store Attribute

Context

There is a number of predefined attributes in the identity store, starting with MX_. You can add user defined attributes in addition to these attributes.

Note

Never start a user defined attribute with MX_.

You can add or modify an attribute in the identity store:

Procedure

1. Select an identity store's [Attributes](#) entry in the tree.
2. Choose the function:
 - To add an attribute, select [Attributes](#) in the tree and choose  [New](#)  [Attribute](#)  from the context menu.
 - To modify an attribute, select the attribute and choose [Properties](#) from the context menu.

Note

The *Identity Store Attribute* dialog box is opened. This dialog box is used to configure attribute properties in the identity store (for all tasks) and to configure some properties specific to one task. Only a limited number of properties can be modified for each task. The values you specify for the attribute are used as default values for the task.

3. Select a sub-tab to enter information about the attribute:

- *General*
- *Storage*
- *Presentation*
- *Assignments* (for assignment attributes only)
- *Validation*
- *Event Processes*
- *Attribute Values*
- *Entry Types*

4. Choose **File > Save** to save the attribute properties.

Related Information

[Attribute Properties \[page 987\]](#)

[General \(Attribute\) \[page 988\]](#)

[Storage \(Attribute\) \[page 989\]](#)

[Presentation \(Attribute\) \[page 992\]](#)

[Assignments \(Attribute\) \[page 996\]](#)

[Validation \(Attribute\) \[page 998\]](#)

[Event Processes \(Attribute\) \[page 999\]](#)

[Attribute Values \(Attribute\) \[page 999\]](#)

[Entry Types \(Attribute\) \[page 1001\]](#)

[Identity Store Attributes \[page 76\]](#)

3.1.6.4.1 Configuring Attribute Ownership and Conflict Handling

Gives an overview of how to configure attribute ownership, conflict handling and how to log conflicts.

Context

You can configure which repositories are allowed to update the attribute in the identity store (attribute ownership). Also you can define to log precedence conflicts to the system log and how they should be handled for the attribute. Do the following:

Procedure

1. Select an identity store's *Attributes* entry in the tree.
2. Select the attribute and choose *Properties* from the context menu.
3. On the *General* tab, configure the following fields:
 - a. *Owner repository*: Select a repository from the list that should be owner for the attribute, or select *None* if no owner should be specified.
 - b. *Conflict handling*: Select a conflict handling method from the list.
 - c. *Log conflicts to system log*: Select the option to log conflicts to system log.
4. Choose  *File*  to save the properties.

Related Information

[Attribute Properties \[page 987\]](#)

[General \(Attribute\) \[page 988\]](#)

[Adding or Modifying an Identity Store Attribute \[page 173\]](#)

[Defining the Storage Parameters of an Identity Store Attribute \[page 176\]](#)

[Configuring the Web Presentation of an Identity Store Attribute \[page 176\]](#)

[Configuring the Web Presentation of an Assignment Attribute \[page 177\]](#)

[Defining the Input Validation for an Identity Store Attribute \[page 178\]](#)

[Defining Event Processes for an Identity Store Attribute \[page 179\]](#)

[Defining Attribute Values for an Identity Store Attribute \[page 180\]](#)

[Defining Entry Types for an Identity Store Attribute \[page 185\]](#)

[Attribute Ownership \[page 47\]](#)

3.1.6.4.2 Defining the Storage Parameters of an Identity Store Attribute

Context

To define how an attribute is stored in the identity store, do the following:

Procedure

1. Select an identity store's *Attributes* entry in the tree.
2. Select the attribute and choose *Properties* from the context menu.
3. Fill in the fields on the *Storage* tab.
4. Choose  *File*  to save the properties.

Related Information

[Storage \(Attribute\) \[page 989\]](#)

[Identity Store Attributes \[page 76\]](#)

3.1.6.4.3 Configuring the Web Presentation of an Identity Store Attribute

Context

Parameters for web presentation of an attribute, like display name, tooltip text or type of input field (presentation), can be specified in the *Presentation* tab. This specifies how the attribute is presented in the web user interface Identity Management User Interface. Web presentation of an assignment attribute is configured in the *Assignments* tab and described in topic [Configuring the Web Presentation of an Assignment Attribute](#).

To configure the web presentation for an attribute, do the following:

Procedure

1. Select an identity store's [Attributes](#) entry in the tree.
2. Select the attribute and choose [Properties](#) from the context menu.
3. Define the web presentation parameters in the [Presentation](#) tab.
4. Choose  [File](#)  to save the properties.

Related Information

[Attribute Properties \[page 987\]](#)

[Presentation \(Attribute\) \[page 992\]](#)

[Configuring the Web Presentation of an Assignment Attribute \[page 177\]](#)

[Assignments \(Attribute\) \[page 996\]](#)

[Identity Store Attributes \[page 76\]](#)

3.1.6.4.4 Configuring the Web Presentation of an Assignment Attribute

Gives an overview of how to configure web presentation of identity store assignment attributes.

Context

Note

Configuration of the web presentation of other identity store attributes is described in topic *Configuring the Web Presentation of an Identity Store Attribute* and is done in the [Presentation](#) tab.

To configure the web presentation of an identity store assignment attribute:

Procedure

1. Select an identity store's [Attributes](#) entry in the tree.
2. Select the attribute and choose [Properties](#) from the context menu.

3. Define the web presentation parameters in the *Assignments* tab.
4. Choose ► *File* ► *Save* ► to save the properties.

Related Information

[Attribute Properties \[page 987\]](#)

[Assignments \(Attribute\) \[page 996\]](#)

[Configuring the Web Presentation of an Identity Store Attribute \[page 176\]](#)

[Presentation \(Attribute\) \[page 992\]](#)

[Identity Store Attributes \[page 76\]](#)

3.1.6.4.5 Defining the Input Validation for an Identity Store Attribute

Context

To define the rules for the input validation for an identity store attribute, do the following:

Procedure

1. Select an identity store's *Attributes* entry in the tree.
2. Select the attribute and choose *Properties* from the context menu.
3. Define the parameters for input validation in the *Validation* tab.
4. Choose ► *File* ► *Save* ► to save the properties.

Related Information

[Attribute Properties \[page 987\]](#)

[Validation \(Attribute\) \[page 998\]](#)

[Identity Store Attributes \[page 76\]](#)

3.1.6.4.6 Defining Event Processes for an Identity Store Attribute

Context

When configuring an attribute, you use the [Event Processes](#) tab to configure the event processes of the attribute.

Note

Adding event processes for an attribute can affect performance, and should only be used if strictly necessary.

To define event processes for an identity store attribute, proceed as follows:

Procedure

1. Select an identity store's [Attributes](#) entry in the tree.
2. Select the attribute and choose [Open](#) from the context menu.
3. On the [Event Processes](#) tab, define add, modify and/or delete event processes for the attribute. Use the ... button to the right of the field to select and add the event process. You can perform the following operations:
 - For the [Add](#) a process operation, you can only define a process of type [Add Attribute Process](#).
 - For the [Modify](#) a process operation, you can only define a process of type [Modify Attribute Process](#).
 - For the [Delete](#) a process operation, you can only define a process of type [Delete Attribute Process](#).

Note

After defining adding, modifying and/or deleting event processes for the attribute, you need to make the processes public.

4. Choose  [File](#)  [Save](#)  to save the properties.

Related Information

[Attribute Properties \[page 987\]](#)

[Event Processes \(Attribute\) \[page 999\]](#)

[Identity Store Attributes \[page 76\]](#)

3.1.6.4.7 Defining Attribute Values for an Identity Store Attribute

Context

To define event tasks for an identity store attribute, do the following:

Procedure

1. Select an identity store's *Attributes* entry in the tree.
2. Select the attribute and choose *Properties* from the context menu.
3. Define attribute values for the attribute in the *Attribute values* tab.
4. Choose  *File*  to save the properties.

Related Information

[Attribute Properties \[page 987\]](#)

[Attribute Values \(Attribute\) \[page 999\]](#)

[Defining an SQL Query to Create a List of Legal Values \[page 180\]](#)

[Defining an SQL Query to Create a Value Dependent List of Legal Values \[page 182\]](#)

[Defining Attribute Value Help \[page 183\]](#)

[Identity Store Attributes \[page 76\]](#)

3.1.6.4.7.1 Defining an SQL Query to Create a List of Legal Values

You can specify an SQL query that produces a list of legal values. This list will be dynamic, depending on the result of the SQL query. The query normally returns a list of unique values, for instance:

```
SELECT distinct avalue FROM idmv_value_basic WHERE attrname='MX_LOCATION'
```

Note

All variables and constants are enclosed in a % pair (%name%). If there is a % sign in the output, you must use the keyword \$PERCENT, for instance:

```
SELECT <columns> FROM <table> WITH (NOLOCK) WHERE <column>=%ADMINMSKEY% AND  
<column> like '<value>$PERCENT'
```

```
SELECT ValText FROM mxi_AttrValueHelp WHERE ValId = %ADMINMSKEY% AND Valkey  
LIKE 'DEV$PERCENT'
```

Note

Make sure that the SQL statement is efficient, also if the amount of data is high.

If you retrieve values from the identity store, it is recommended to use one of the views installed with the product.

Note

When using Microsoft SQL Server, it is recommended that all SELECT queries should include "with (nolock)". This will reduce the table locking, improve performance, and reduce the probability of deadlocks. For instance:

```
SELECT <columns> FROM <table> WITH (NOLOCK) WHERE <column>='<value>'
```

Note

SQL query with "order by" keyword is not supported for attributes with *SingleLine* presentation type. If you need the results to be ordered (that is, using SQL query with "order by" keyword), you can use attributes with *SingleSelect* presentation type.

Related Information

[Adding or Modifying an Identity Store Attribute \[page 173\]](#)

[Defining Attribute Values for an Identity Store Attribute \[page 180\]](#)

[Defining an SQL Query to Create a Value Dependent List of Legal Values \[page 182\]](#)

[Defining Attribute Value Help \[page 183\]](#)

[Attribute Values \(Attribute\) \[page 999\]](#)

[Attribute Properties \[page 987\]](#)

3.1.6.4.7.2 Defining an SQL Query to Create a Value Dependent List of Legal Values

You can also create a list of legal values for one attribute that depends on the selected value in another attribute. For instance, if you have two fields in the Identity Management User Interface, "Country" and "City", the list of legal values for "City" depends on which country the user has selected in the "Country" field.

Note

This should be configured as part of the attribute definition on a task, as it depends on both attributes being present in the Identity Management User Interface.

Note

Both fields must be defined as "Single value" with presentation type "Single select".

You use the syntax in the WHERE clause of the SQL query:

```
%FIELD.<ATTRIBUTE_NAME>%
```

To create the list of legal values for the MX_ADDRESS_CITY attribute, you define an SQL query for this attribute that may look like this (for Microsoft SQL Server):

```
SELECT City from MyCities with (nolock) where Country =  
%FIELD.MX_ADDRESS_COUNTRY%
```

Note

Make sure not to enclose %FIELD.MX_ADDRESS_COUNTRY% in apostrophes.

The list of cities is stored in a separate table, MyCities. City and Country in this example must be columns in this table.

You also have to define a list of legal values for the MX_ADDRESS_COUNTRY that the user can select from. Make sure that all possible values for MX_ADDRESS_COUNTRY are present in MyCities and that there are values for City corresponding to each value for Country.

Legal values for MX_ADDRESS_CITY are only displayed after the user has selected a value for MX_ADDRESS_COUNTRY.

Related Information

[Adding or Modifying an Identity Store Attribute \[page 173\]](#)

[Defining Attribute Values for an Identity Store Attribute \[page 180\]](#)

[Defining an SQL Query to Create a List of Legal Values \[page 180\]](#)

[Defining Attribute Value Help \[page 183\]](#)

[Attribute Values \(Attribute\) \[page 999\]](#)

[Attribute Properties \[page 987\]](#)

3.1.6.4.7.3 Defining Attribute Value Help

When you specify legal values for an attribute, you can configure attribute value help for the attribute. This gives the possibility to provide language dependent value help. As opposed to the other mechanisms, it is the key of the value help that is stored as the attribute value, and not the value.

Presentation Types

Attribute value help can be combined with several presentation types:

- [SingleSelect](#)
The values are displayed in a drop-down list.
- [Radio button](#)
The values are displayed a selectable options.
- [ObjectValueHelp](#)
The values are displayed in a separate window where the user can search for specific values.
- [MultiSelect](#)
The values are displayed as drop-down lists for each value in a multi-value attribute.

The mxi_AttrValueHelp Table

The value help is stored in a table in the Identity Center database. The default table is mxi_AttrValueHelp. This table has the following structure:

Column name	Data type	Description
ValId	varchar(80), not null	Value group identifier. This is needed to be able to keep values for multiple attributes in the same table.
ValKey	nvarchar(80), not null	The key to be stored in the database.
ValLocale	varchar(16)	This is the language definition on the following format: <Language>_<Country>_<Variant>
ValText	nvarchar(512), not null	This is the text which will appear in the list in the User Interface.

There are some pre-installed values in this table. New values can be inserted into this table using a job that reads values from a given repository, a file or some other source.

It is also possible to create a separate table for this. This can be necessary if the data set is large to avoid degraded performance.

The access rights must be defined as follows:

- The <prefix>_rt user must have insert/update/delete access if you want the runtime engine to populate the table.
- The <prefix>_prov users must have select access.

Example

Valid	ValKey	ValLocale	ValText
SALUT	SAL_KEY_1	DE_DE	Herr
SALUT	SAL_KEY_1	EN	Mr
SALUT	SAL_KEY_1	NO_NO	Hr
SALUT	SAL_KEY_2	DE_DE	Frau
SALUT	SAL_KEY_2	EN	Mrs
SALUT	SAL_KEY_2	NO_NO	Fru
SEX	SEX_M	EN	Male
SEX	SEX_M	NO_NO	Mann
SEX	SEX_F	EN	Female
SEX	SEX_F	NO_NO	Kvinne

You can choose if you want to include language dependent values or not. If you leave the "ValLocale" column empty and only use one value in the "ValText" column for each text in the "ValKey" column.

User Interface

When a user asks for value help for an attribute, the user's language setting (retrieved from UME) is used to determine which values to display. The values matching the user's language setting are displayed. If <Variant> is not found, <Language>_<Country> is used. If <Country> is not found, <Language> is used. If there is no key for the requested language the values for English are displayed.

To Identity Store pass

When updating an attribute where value help is defined, it is the key that is written to the attribute. This key is verified against the list of legal values and if the key is not found, the whole entry fails.

Related Information

[Adding or Modifying an Identity Store Attribute \[page 173\]](#)

[Defining Attribute Values for an Identity Store Attribute \[page 180\]](#)

[Defining an SQL Query to Create a List of Legal Values \[page 180\]](#)

[Defining an SQL Query to Create a Value Dependent List of Legal Values \[page 182\]](#)

[Attribute Properties \[page 987\]](#)

[Attribute Values \(Attribute\) \[page 999\]](#)

3.1.6.4.8 Defining Entry Types for an Identity Store Attribute

Context

To define entry types in which this attribute is included:

Procedure

1. Select an identity store's *Attributes* entry in the tree.
2. Select the attribute and choose *Properties* from the context menu.
3. Select the *Entry types* tab.
4. Define the entry types for the attribute. The list contains all defined entry types. For each entry type you can specify whether or not the attribute is allowed or if it is mandatory. Specify the following parameters:
 - *Allow*
 - *Mandatory*
 - *List*
 - *Ext. properties*
5. Choose  *File*  *Save*  to save the properties.

Related Information

[Attribute Properties \[page 987\]](#)

[Entry Types \(Attribute\) \[page 1001\]](#)

[Identity Store Attributes \[page 76\]](#)

3.1.6.4.8.1 Configuring Extended Attribute Properties

Context

When you configure attributes for an entry type or entry types for an attribute, you use the [Extended attribute properties](#) dialog box to configure properties for multivalued reference attributes, like roles or privileges. You decide which information should be available in the Identity Management User Interface when viewing or modifying an assignment.

Note

If you have used the Assignment request task to create the assignment, make sure that the parameters of that task match this configuration.

Procedure

1. View the properties of an entry type and select the [Attributes](#) tab or view the properties of an attribute and select the [Entry types](#) tab.
2. Select the ... button behind one of the multivalued reference attributes, for instance MXREF_MX_ROLE to open the [Extended attribute properties](#) dialog box.
3. Fill in the fields in the dialog box.
4. Choose [OK](#) to close the dialog box and to save the properties.

Related Information

[Defining Entry Types for an Identity Store Attribute \[page 185\]](#)

[Entry Types \(Attribute\) \[page 1001\]](#)

[Extended Attribute Properties \[page 985\]](#)

[Attributes \(Entry Type\) \[page 985\]](#)

[Defining Attributes for an Entry Type \[page 171\]](#)

3.1.6.5 Importing Identity Store Schema

Context

You can import the schema of an identity store.

Procedure

1. In the tree view, expand your identity store and select the *Identity Store Schema* node.
2. Choose *Import* from the context menu.
3. Locate and select the file containing the schema you want to import.
4. Decide whether you want the new schema to override the current schema.

Import Context	Selected Option	
	Yes	No
The new schema includes attributes and entry types which are not present in the current schema	New attributes and entry types will be imported in the target identity store.	New attributes and entry types will be imported in the target identity store.
The same attributes and entry types exist in both schemas - the current and the new one	Attributes and entry types, along with configurations of attributes for a specific entry type, from the new schema will overwrite the existing ones.	Attributes and entry types, along with configurations of attributes for a specific entry type, from the current schema will be preserved.
	Example 1: In the new schema you have changed the <i>Display Name</i> of the MX_PERSON entry type. When the schema is imported, the value of the <i>Display Name</i> is overwritten by the value from the new schema.	Example 1: In the new schema you have changed the <i>Display Name</i> of the MX_PERSON entry type. When the schema is imported, the value of the <i>Display Name</i> from the current schema is preserved.
	Example 2: In the new schema you have changed the MX_ADDRESS_CITY attribute in the MX_PERSON entry type to be mandatory. When the schema is imported, the configuration is overwritten and the MX_ADDRESS_CITY is set as mandatory for MX_PERSON.	Example 2: In the new schema you have changed the MX_ADDRESS_CITY attribute in the MX_PERSON entry type to be mandatory. When the schema is imported, the configuration is preserved and MX_ADDRESS_CITY remains as it was in the current identity store schema.

Import Context	Selected Option	
	Yes	No
The current schema includes attributes and entry types which are not present in the new schema	Deleted attributes or entry types from the new schema will not be deleted from the current schema after the import.	Deleted attributes or entry types from the new schema will not be deleted from the current schema after the import.

3.1.6.6 Exporting Identity Store Schema

Context

You can export the schema of an identity store with all entry types and attributes.

Procedure

1. In the tree view, expand your identity store and select the *Identity Store Schema* node.
2. Choose *Export* from the context menu.
3. Select a location and enter a file name for the export file.
4. Make sure the Identity Store Schema File is selected in the *Save as type* field.
5. Choose *Save*.

❗ Note

Some properties for the entry types and attributes are not exported in the Identity Store Schema File. They are as follows:

- Entry types:
 - Display form
 - Search form
 - All event processes
- Attributes:
 - All event processes

Those properties are part of the package import/export logic and will be exported/imported during package transport.

3.1.7 Maintaining Packages

3.1.7.1 Creating Package Folders

You can create package folders to help structure the packages in the Identity Center.

Context

To add a package folder, proceed as follows:

Procedure

1. In the tree view, select the package folder where you want to add the sub-folder and choose ► [New Package Folder](#) from the context menu.
2. Enter a name for the package folder.

Results

The package folder is added, and you can now add packages to this package folder.

Related Information

[Creating Packages \[page 190\]](#)

[Package Folder Overview \[page 1003\]](#)

3.1.7.2 Viewing Packages in Package Folder

You can get an overview with information about all packages within a package folder and all its sub-folders.

Context

To view the package list, proceed as follows:

Procedure

1. Select a package folder in the tree view and choose [Properties](#) from the context menu.
2. The table shows properties of all packages in the folder and sub-folders.

Related Information

[Package Folder Overview \[page 1003\]](#)

[Creating Package Folders \[page 189\]](#)

3.1.7.3 Creating Packages

You can add packages that contain parts of the configuration that can be delivered as a whole.

Prerequisites

- You must be an [Admin](#) user to be able to create a package.

Context

To add a package, proceed as follows:

Procedure

1. Select the package folder where you want to add the package and choose **► New ► Package ►** from the context menu.
2. Enter a name for the package. Legal characters are *a-z*, *A-Z*, *0-9*, *_* (underscore) and *.* (period).
The first part is the qualified name that was provided when the Identity Management database was installed.
3. Choose *OK*.

Results

The package is added to the tree view and is locked by you. You can start adding your configuration to the package.

3.1.7.4 Viewing and Configuring Package Properties

Context

To view package properties, do the following:

Procedure

1. Select the package in the tree view and choose *Properties* from the context menu.
2. Select a tab to enter or view the information about the package:
 - *General*
 - *Public Objects*
 - *Dependencies*
 - *Authorized Users*
 - *Revision History*

3. Choose  [File](#)  to save any changes made to the package properties.

Related Information

[Package Properties \[page 1003\]](#)

[General \(Packages\) \[page 1004\]](#)

[Public Objects \(Packages\) \[page 1005\]](#)

[Dependencies \(Packages\) \[page 1005\]](#)

[Authorized Users \(Packages\) \[page 1009\]](#)

[Revision History \(Packages\) \[page 1010\]](#)

3.1.7.5 Viewing General Properties of a Package

View the general properties of a package to see information about who has locked it, which version it is, and when it was last modified.

Context

To view general information about the package, proceed as follows:

Procedure

1. Select the package in the tree view.
2. Choose [Properties](#) from the context menu to open the package properties.
3. Select the [General](#) tab.
4. View the information about the package or add a description of the package.
5. Choose  [File](#)  to save the description.

Related Information

[General \(Packages\) \[page 1004\]](#)

3.1.7.6 Viewing Public Objects in a Package

The [Public Objects](#) tab shows all public objects defined in the package. These objects are available from other packages.

Context

To view the public objects in the package, proceed as follows:

Procedure

1. Select the package in the tree view.
2. Choose [Properties](#) from the context menu to open the package properties.
3. Select the [Public Objects](#) tab.
4. The list contains information about all public objects in the package.

Related Information

[Public Objects \(Packages\) \[page 1005\]](#)

3.1.7.7 Viewing External Dependencies of a Package

The [Dependencies](#) tab shows all external dependencies to and from this package.

Context

To view the package external dependencies, proceed as follows:

Procedure

1. Select the package in the tree view.
2. Choose [Properties](#) from the context menu.
3. Select the [Dependencies](#) tab.

4. The two tables - *Outgoing* and *Incoming*, show dependencies to and from the package.

Note

If a job in a package contains job variables of type *Process Reference*, the referenced public processes from other packages are not displayed on the *Dependencies* tab of the package. Job variables are stored in a <job name>.ddx file in the same directory as the job file.

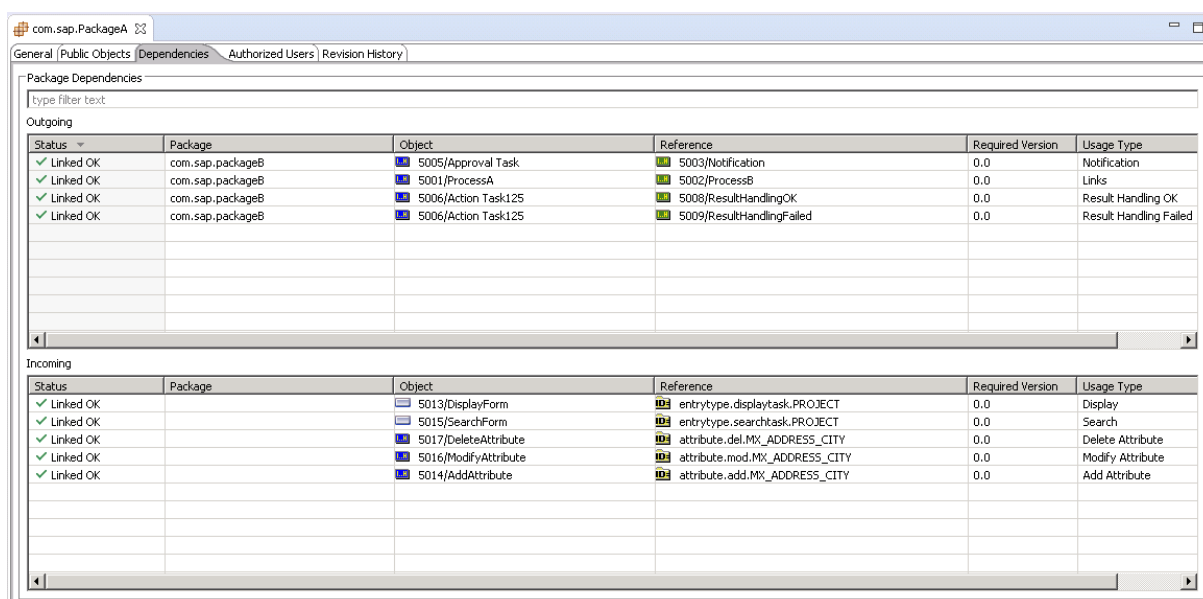
Example

You have package A that contains the following public objects: processes (Process A, Add Attribute, Modify Attribute and Delete Attribute) and forms (Display and Search). You have package B that contains the following public processes: Process B, Notification, Result Handling OK, Result Handling Failed.

The example below shows cross-package dependencies between package A and package B, as well as identity store schema (entry types and attributes) dependencies.

- *Outgoing* dependencies in package A
Shows all references from this package (package A) to other packages (package B).
 - Process A contains a reference (link) to process B.
 - Approval task in process A contains a reference to notification process in package B.
 - Action task in process A contains references to Result Handling OK process and Result Handling Failed process in package B.
- *Incoming* dependencies in package A:
 - Display and Search forms are defined on entry type Project.
 - Event processes Add Attribute, Modify Attribute and Delete Attribute are defined on MX_ADDRESS_CITY attribute.

Event processes that are defined on entry types and attributes are always displayed as incoming dependencies.



Status	Package	Object	Reference	Required Version	Usage Type
✓ Linked OK	com.sap.packageB	5005/Approval Task	5003/Notification	0.0	Notification
✓ Linked OK	com.sap.packageB	5001/ProcessA	5002/ProcessB	0.0	Links
✓ Linked OK	com.sap.packageB	5006/Action Task125	5008/ResultHandlingOK	0.0	Result Handling OK
✓ Linked OK	com.sap.packageB	5006/Action Task125	5009/ResultHandlingFailed	0.0	Result Handling Failed

Status	Package	Object	Reference	Required Version	Usage Type
✓ Linked OK		5013/DisplayForm	entrytype.displaytask.PROJECT	0.0	Display
✓ Linked OK		5015/SearchForm	entrytype.searchtask.PROJECT	0.0	Search
✓ Linked OK		5017/DeleteAttribute	attribute.del.MX_ADDRESS_CITY	0.0	Delete Attribute
✓ Linked OK		5016/ModifyAttribute	attribute.mod.MX_ADDRESS_CITY	0.0	Modify Attribute
✓ Linked OK		5014/AddAttribute	attribute.add.MX_ADDRESS_CITY	0.0	Add Attribute

- **Incoming** dependencies in package B:
Shows all references from other packages (package A) to public objects in this package (package B).

The screenshot shows the 'Package Dependencies' window with the 'Incoming' tab selected. The window title bar shows 'com.sap.PackageA' and 'com.sap.packageB'. The 'Incoming' section displays a table of dependencies from package A to package B.

Status	Package	Object	Reference	Required Version	Usage Type
✓ Linked OK	com.sap.PackageA	5005/Approval Task	5003/Notification	0.0	Notification
✓ Linked OK	com.sap.PackageA	5001/ProcessA	5002/ProcessB	0.0	Links
✓ Linked OK	com.sap.PackageA	5006/Action Task125	5008/ResultHandlingOK	0.0	Result Handling OK
✓ Linked OK	com.sap.PackageA	5006/Action Task125	5009/ResultHandlingFailed	0.0	Result Handling Failed

Related Information

[Dependencies \(Packages\) \[page 1005\]](#)

3.1.7.8 Viewing Authorized Users of a Package

The **Authorized Users** tab shows all users that have access to the package and which type of access they have.

Context

To view the authorized users of a package, proceed as follows:

Procedure

1. Select the package in the tree view.
2. Choose **Properties** from the context menu to open the package properties.
3. Select the **Authorized Users** tab.
4. The list shows all users who have access to the package.

Related Information

[Authorized Users \(Packages\) \[page 1009\]](#)

3.1.7.9 Providing Authorizations to a Package

You can add users with different authorizations to the package.

Prerequisites

- The users that you want to give access to must be defined in UME and added to the Identity Center database.
- The package must be checked in.
- You must be a package owner to provide access to the package.

Context

To provide access to a package:

Procedure

1. Select the package in the tree view.
2. Choose [Properties](#) from the context menu to open the package properties.
3. Select the [Authorized Users](#) tab.
4. Choose [Add Users...](#) to open the [Add Users](#) dialog box.
5. Select one or more users in the list.
6. Choose [OK](#) to close the dialog box.

The selected users are added to the list.

7. By default, all users are given [View](#) access to the package. Add more authorizations to the users if needed
8. Choose  [File](#)  [Save](#)  to save the change.

Related Information

[Authorized Users \(Packages\) \[page 1009\]](#)

[Viewing Authorized Users of a Package \[page 195\]](#)

3.1.7.10 Removing Authorization to a Package

You can remove users from the list of authorized users.

Prerequisites

- The package must be checked out.
- You must be a package owner to give access to the package.

Context

To remove access to a package:

Procedure

1. Select the package in the tree view.
2. Choose [Properties](#) from the context menu to open the package properties.
3. Select the [Authorized Users](#) tab.
4. Select the user you want to remove and choose [Remove User](#).
5. Choose  [File](#)  [Save](#)  to save the change.

Related Information

[Authorized Users \(Packages\) \[page 1009\]](#)

[Viewing Authorized Users of a Package \[page 195\]](#)

3.1.7.11 Viewing Revision History of a Package

The *Revision History* tab provides an overview of all logged changes to the package.

Context

To view the revision history of a package, proceed as follows:

Procedure

1. Select the package in the tree view.
2. Choose *Properties* from the context menu to open the package properties.
3. Select the *Revision History* tab.
4. The list shows all logged changes to the package.

Related Information

[Revision History \(Packages\) \[page 1010\]](#)

3.1.7.12 Restoring Version of a Package

You can restore a previous version of a package to revert changes done in later versions.

Prerequisites

- The package must be locked to be able to restore.
- You must be a package owner or a developer to be able to restore.
- The action of the version you restore must be one of the following:
 - *Package checked in*
 - *Package imported*
 - *Package restored*

Context

Note

You can also revert a package to the initial state of its current version. For more information, see *Reverting a Package*.

To restore a previous version of a package, proceed as follows:

Procedure

1. Select the package in the tree view.
2. Choose *Properties* from the context menu to open the package properties.
3. Select the *Revision History* tab.
4. Select a restorable version in the list and choose *Restore Selected*.

Note

This will overwrite the existing package contents with the contents of the selected version. You can expect the same behavior as when importing a package from a source to a target system, described in: [Importing Packages \[page 201\]](#).

Related Information

[Viewing Revision History of a Package \[page 198\]](#)

[Revision History \(Packages\) \[page 1010\]](#)

[Reverting Packages \[page 203\]](#)

3.1.7.13 Checking Out Packages

To modify a package, you need to check out (lock) the package.

Context

By default, only one user can check out a package and work on its configuration. If explicitly required, you can change this so that multiple users to be able to check out one and the same package and work simultaneously on its configuration. See *Configuring the Java System Properties* in the *Installing SAP Identity Management Developer Studio* section.

Note

We do not recommend multiple users to check out one and the same package. If multiple users modify one and the same package object, (for example, a process) they will override each other's changes.

Note

The first user who checks in the package, checks in (saves) the changes of all the users that checked out this package.

To check out a package, proceed as follows:

Procedure

1. In the Identity Management Developer Studio tree view, select the package that you want to check out.
2. Choose [Check out](#) from the context menu.
3. The package is displayed with a green padlock in the tree view.

Note

All packages that you have checked out have green padlocks. Packages that have been checked out by other users have red padlocks. Packages that have been checked out simultaneously by you and other users have green padlocks.

Related Information

[Configuring the Java System Properties](#)

3.1.7.14 Checking In Packages

To save changes when modifying a package, you need to check in the package.

Context

Note

If multiple users have checked out one and the same package and work on its configuration, the first user who checks in the package, checks in (saves) the changes of all the users.

To check in a package, proceed as follows:

Procedure

1. Select the package in the tree view.
2. Choose *Check in* from the context menu.

You can add a description of the changes when checking in.

3. Choose OK

Note

The package is displayed without a padlock icon in the tree view.

3.1.7.15 Importing Packages

You can use the import and export functionality of a package to transport your configuration.

Context

Transport of configuration is done on a package level. Each package is transported separately.

You can import a package with cross-package dependencies, for example, a package that contains references to processes or scripts in other packages. In order for the references to work, you need to have all dependent packages imported.

Note

When you import a package from one system into another (for example, from a test system into a productive system), it is recommended to have identical dispatcher names in both systems. The same applies for the referenced identity stores. If the names are not unified, this may cause some runtime functional issues.

To avoid issues during package import related to missing references, a warning dialog will appear that will provide information about missing references related to event processes/forms, dispatches, and identity stores. All warning messages are also written in the NWA logs. For more information, see [3194957](#) .

When importing a package from a source to a target system, expect the following behavior:

- If the package does not exist in the target system, the package content is imported as-is.
- If the package already exists in the target system:
 - New package objects in the source system are imported into the target system.
 - Modified package objects in the source system, such as constant values, are imported into the target system. Modified values overwrite the existing ones with the following exception: constants of types

other than **Job Reference** and **Process Reference** that have *Show in the Admin User Interface* enabled.

For example: If a **String** constant and a **Job Reference** constant are modified (both have *Show in the Admin User Interface* enabled), after package import, the value of the **String** constant will not be overwritten, while the value of the **Job Reference** constant will be overwritten.

- Deleted package objects in the source system, such as repository types and message templates, are preserved in the target system. If you want to delete them in the target system, you need to do it manually. Alternatively, you can delete the package in the target system and import the package from the source system.

Note

You cannot import a package with the same name into two different identity stores.

When a new version of a package is imported, jobs and tasks running at the time of the import and those being scheduled for processing in the provisioning queue before the package import, become obsolete. For more information, see [Obsolescence \[page 87\]](#).

To import a package, proceed as follows:

Procedure

1. In the Identity Management Developer Studio tree view, select the identity store where you want to import the package to.
2. Select the **Packages** folder and choose *Import* from the context menu.
3. Navigate to the package file that you want to import, select it and choose *Open*.
4. In the *Enter import reason* dialog, configure the following:
 - (Optionally) Enter a reason for the import.
 - Choose one of the following:
 - Choose *Import*.
The package is imported with its package qualified name.
 - Choose *Import as new package* and enter a name for the package.
The package is imported under a new name. You can use this option if you want to modify a configuration package and keep your changes when updating the respective framework.
5. Choose *OK*.

Related Information

[Packages \[page 91\]](#)

3.1.7.16 Exporting Packages

You can use the import and export functionality of a package to transport your configuration.

Prerequisites

The package must be checked in.

Context

Transport of configuration is done on a package level. Each package is transported separately.

Note

When you transport a package, note the following:

- Job variables are not exported with the package.
- If the package contains an approval task with approvers defined on the task itself, the approvers are exported with their MSKEYVALUE and not with their MSKEY.
- The schedule rule you have set for a repository job is not exported and imported with the package. In this case, the repository job is imported with its default schedule rule "On demand".

To export a package, proceed as follows:

Procedure

1. In the tree view, select the package that you want to export.
2. Choose [Export](#) from the context menu.
3. Navigate to the location where you want to export the package to and choose [Save](#).

3.1.7.17 Reverting Packages

You can revert a package to the initial state of its current version.

Prerequisites

- The package is checked out.

- You must be a package owner or a developer.

Context

Note

You can also restore a previous version of the package. For more information, see *Restoring Version of a Package*.

To revert a package to the initial state of its current version, proceed as follows:

Procedure

1. Select the package in the tree view.
2. Choose *Revert Package...* from the context menu.
3. Choose *OK*.

Note

When reverting the package, all changes done to the current version of the package are discarded. You can expect the same behavior as when importing a package from a source to a target system, described in: [Importing Packages \[page 201\]](#).

Related Information

[Restoring Version of a Package \[page 198\]](#)

3.1.7.18 Copying Packages

You can copy packages within an identity store.

Prerequisites

- You must at least have view authorization for the package.

Context

To copy a package, proceed as follows:

Procedure

1. Select the package in the tree view.
2. Choose [Copy](#) from the context menu.
3. Select the package folder where you want to create the copy.
4. Choose [Paste](#) from the context menu.
5. Enter a new name for the package in the dialog box and choose [OK](#).

Results

The package is copied. If objects in the copied package contain references to processes in other packages, those references are preserved.

3.1.7.19 Moving Packages

You can move (cut) a package from one package folder to another within the same identity store.

Prerequisites

- You have checked out the package.

Context

To move a package, you can drag-and-drop it to the new package folder. If your package is checked out, you can also use the cut-and-paste function described in the procedure below.

To move a package, proceed as follows:

Procedure

1. Select the package in the tree view.
2. Choose *Cut* from the context menu.
3. Select the package folder that you want to move the package to.
4. Choose *Paste* from the context menu.

You can also move the package using drag-and-drop.

Results

The package is moved to the selected package folder. The qualified name, authorizations and contents remain the same.

3.1.7.20 Renaming Packages

You can give a package a new name.

Prerequisites

- You have checked out the package.

Context

Note

When a package is renamed, all references to public objects in this package will be invalidated. The references will be restored if the name of the package is changed back again, or a package with the same qualified name is imported. Otherwise, references have to be restored manually.

To rename a package, proceed as follows:

Procedure

1. Select the package in the tree view.
2. Choose [Rename](#) from the context menu.
3. Enter a new qualified name for the package in the tree view.

Results

The package is renamed and has a new qualified name. The authorizations and contents of the package are not changed.

Next Steps

Any references to public objects in this package have to be updated.

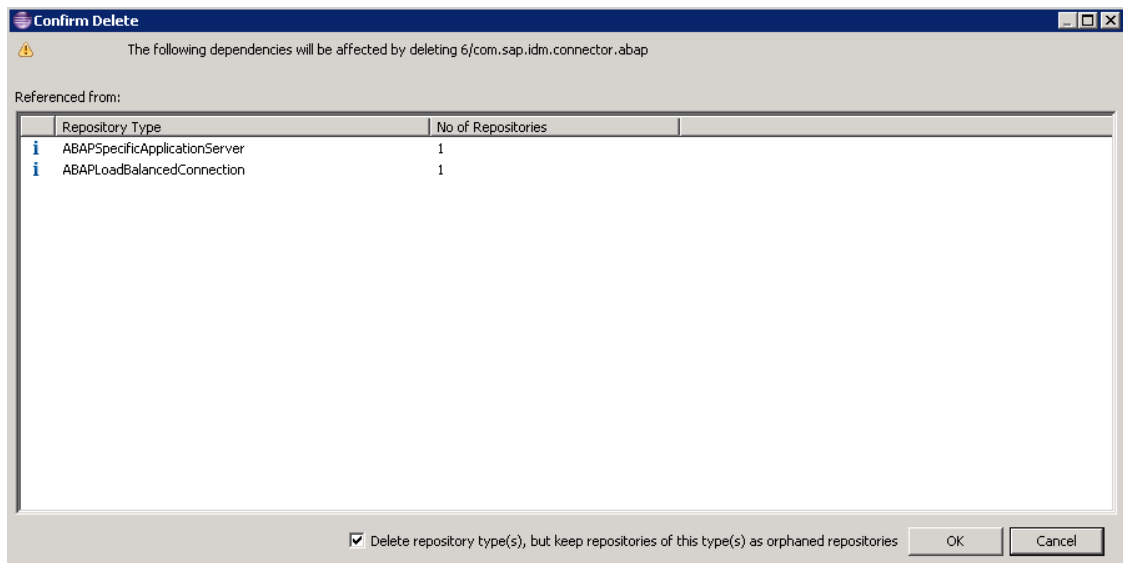
3.1.7.21 Deleting Packages

Context

To delete a package, proceed as follows:

Procedure

1. Select the package and choose [Delete](#) from the context menu.
2. If you don't have repositories based on repository type(s) of this package, choose [Yes](#) to delete the package.
3. If you have repositories based on repository type(s) of this package, in [Confirm Delete](#) dialog, decide how to proceed with the repositories.
 - To delete repository type(s), but keep repositories of this type(s) as orphaned repositories, select the checkbox at the bottom of the dialog and choose [OK](#).



Note

You can view the orphaned repositories in the Identity Management Administration User Interface but you cannot modify them. In addition, you cannot create and run repository jobs for orphaned repositories and you cannot select these repositories for a job, a pass and a process, for example.

- To delete the repository type(s) and repositories of this type(s), deselect the checkbox at the bottom of the dialog and choose [OK](#).

3.1.7.22 Adding Package Constants

You add package constants to control the behavior of the package. When package constants are defined for a package, administrators are able to view this package in SAP Identity Management Administration User Interface and modify its package constants.

Prerequisites

- You have checked out the package.

Context

To add a package constant, proceed as follows:

Procedure

1. In the tree view, select the package that you want to add a package constant for.
2. Select [Constants](#) and choose [New Package Constant](#) from the context menu.
3. Fill in the fields in the [Package Constant](#) dialog box.
4. Choose [Save](#).

Related Information

[Constants \(Package\) \[page 1011\]](#)

3.1.7.23 Checking for Package Script Usage

You can check if a certain package script is used in any job, task, or other script in the selected Identity Store.

Context

The search is executed through all job references, all job scripts code, and in all package script code of each package.

Note

The search is conducted among the last checked in version of the packages in the identity store. Hence, you should check in all packages to ensure accuracy.

Example

For example, let's assume that you have:

1. A package with name 'packageA'
2. A package with name 'packageB'
3. A package script with name 'PackageScriptUsedInAllPackages' and it's used from both packages
4. Both packages are checked in

If you use the new option for 'PackageScriptUsedInAllPackages', the search will return two results: 'packageA' and 'packageB'.

Now, check out 'packageA' and remove the script usage – 'PackageScriptUsedInAllPackages'. If you execute the search again, it will again return two results – 'packageA' and 'packageB', because the 'packageA' is not checked in and the search is still using its previous version (the last checked in version). Once you check in 'packageA', the search will return only one result - 'packageB'.

Only the syntactically correct expressions for calling a script are considered during the search. Which means that the search result will include only scripts with the following script syntax: `<script name>(<any number of arguments>)`.

❖ Example

Let's assume that you have the following code in several package scripts:

```
var queryResult = executeQuery(sqlQuery);
var queryResult = executeQuery(sqlQuery, dbType);
// var queryResult = executeQuery(sqlQuery);
// executeQuery();
// executeQuery
```

If you perform a search for package script `executeQuery`, it will find all of the usages above except for the last one, which is not a syntactically correct script calling.

Regardless of the times the searched script is used in some job or a script, it will be displayed only once in the search results.

❖ Example

For example, let's assume that you have the following code in a package script:

```
var getSettingValueQuery = "SELECT settingValue FROM settings where settingID = 1";
    var settingValue = executeQuery(getSettingValueQuery);
    // do something with the retrieved setting value

getSettingValueQuery = "SELECT settingValue FROM settings where settingID = 2";
    settingValue = executeQuery(getSettingValueQuery);
    // do something with the retrieved setting value
```

When you perform a search for package script `executeQuery`, it will be displayed only once in the search results.

The search uses the package script name, which means that any commented code is also considered during the search. It also includes all package script references, regardless of their usage. Therefore, you should check all results to verify that the script is actually used.

To search for a package script usage, execute the following steps:

Procedure

1. Navigate to [Package scripts](#) view in the Identity Management Developer Studio.
2. Select the package script and choose the [Check Where Used](#) option by:

- right-clicking the script and choosing the option from the context menu
- selecting the respective icon on the top right side of the *Package script* view

The results of the executed search are displayed in the *Search* view.

3. If the search result contains a long list, you can limit the list to the entries corresponding to the value you specify in the *Filter* field.
4. You can open the script, by selecting it and choosing *Open* from the context menu.
5. You can locate the script in the configuration, by selecting it and choosing *Locate* from the context menu.
6. Choose *Refresh* from the context menu to perform the search again and update the search result.

3.1.8 Maintaining Repository Types

3.1.8.1 Creating a Repository Type

Context

To create a repository type, do the following:

Procedure

1. Make sure that the package where you want to add the repository type is checked out and ready for editing, then select the *Repository types* folder in the tree view.
2. Choose *New repository type* from the context menu.

This opens a repository type dialog box.

3. Select a repository type template.
4. Enter a name for the repository type.
5. Choose *Finish*

The repository type will be added in the tree view and its properties opened for editing. The default list of parameters in the *Parameters* tab will vary depending on the repository type created (which template it is based on).

Next Steps

You create repositories of this type from the Identity Management Administration Interface. For more information, see [SAP Identity Management Solution Operation Guide](#).

Related Information

[Repository Type Properties \[page 1012\]](#)

[Viewing and Configuring Repository Type Properties \[page 212\]](#)

3.1.8.2 Viewing and Configuring Repository Type Properties

Context

To view or configure the repository type properties, perform the following steps in **Identity Management Developer Studio**:

Procedure

1. Select a repository type in the tree view and choose *Open* from the context menu.
2. Select a tab to enter or view the information about the repository type:
 - *General*
 - *Privileges*
 - *Assignment grouping*
 - *Event process*
 - *Context*
3. Choose **File > Save** or press **Ctrl + S** to save any changes made to the repository type properties.

Related Information

[Repository Type Properties \[page 1012\]](#)

[General \(Repository Type\) \[page 1012\]](#)

[Privileges \(Repository Type\) \[page 1014\]](#)

[Assignment Grouping \(Repository Type\) \[page 1015\]](#)

[Event Process \(Repository Type\) \[page 1016\]](#)

[Context \(Repository Type\) \[page 1021\]](#)

3.1.8.3 Defining the Constants for the Repository Type

Context

To define the constants on the repository type, proceed as follows:

Procedure

1. Select the repository type in the tree view and view the properties by choosing *Open* from the context menu.
2. Select the *General* tab.
3. To add the constants to the list, choose *New*.
4. The *Repository Type Properties* dialog box appears where the details for the constant can be defined. Enter the details in the fields.
5. Choose *OK* to close the dialog box and add the constant to the list. To add more constants, repeat steps 1 to 4.

Defining the constants on other tabs in the repository type properties (the tabs *Privileges*, *Assignment grouping*, *Event process* or *Context*), will add those constants to the list of constants in the *General* tab. Their category is *Repository type constant* by default. The category can be changed.

To change the category, or any other value of the constant, select the constant in the list and choose *Modify*. Modify the constant and choose *OK* to add the change to the list.

6. Optionally, move the constant up or down in the list by selecting it and choosing *Up* or *Down*. The order of the constants in this list will determine the order in which the constants are displayed in a repository of this repository type.
7. If a constant needs to be deleted, select the constant in the list and choose *Delete*.
8. Choose  *File*  or press Ctrl + S to save the changes.

Related Information

[Repository Type Properties \[page 1012\]](#)

[General \(Repository Type\) \[page 1012\]](#)

3.1.8.4 Configuring Privilege Policy for a Repository Type

Context

Use the [Privileges](#) tab to configure privilege policy for master privilege:

❗ Note

The master privilege policies are only relevant if the master privilege being assigned has an [add member task](#) defined.

Procedure

1. Select the repository type in the tree view and view the properties by choosing [Open](#) from the context menu.
2. Select the [Privileges](#) tab.
3. Configure the privilege policy options.
4. Choose ► [File](#) ► [Save](#) or press `Ctrl` + `S` to save the changes.

Related Information

[Repository Type Properties \[page 1012\]](#)

[Privileges \(Repository Type\) \[page 1014\]](#)

3.1.8.5 Configuring the Assignment Grouping

Context

Grouping applies to all privileges which are assigned in the same operation. By default, no grouping of assignments is performed in the system, i.e. the member event tasks are executed for every privilege.

To configure the assignment grouping:

Procedure

1. View the properties of the repository type (select the repository type in the tree view and select *Open* from the context menu).
2. Select the *Assignment grouping* tab.
3. Select the *Grouping* option. This means that the member event tasks are executed for a group of assignments and not for each assignment individually.
4. Assignments are separated according to what is selected in this step. Choose between the following options or any combinations of these:
 - *Separate by operation*
 - *Separate by application*
 - *Separate by privilege attribute*, where you also have to select a privilege attribute from the list of available privilege attributes

Note

If none of the options are selected, all assignments will be handled by one execution of the member event tasks.

5. Choose **File > Save** or press **Ctrl** + **S** to save the changes.

Related Information

[Repository Type Properties \[page 1012\]](#)

[Assignment Grouping \(Repository Type\) \[page 1015\]](#)

[Assignment Grouping \[page 57\]](#)

3.1.8.6 Configuring Event Processes for a Repository Type

Context

Use the [Event process](#) tab in the repository type properties to configure event processes for the repository type. Do the following:

Procedure

1. Select the repository type in the tree view and view the properties by choosing [Open](#) from the context menu.
2. Select the [Event process](#) tab.
3. Fill in the necessary fields to configure the event processes for the repository type. Choose ... to navigate to and select the process, then choose [OK](#) to add the process. Define when the processes [validate add](#), [add](#) and [notification of link expiry](#) should be executed in the [Execute](#) list.
4. Choose [File](#) > [Save](#) or press `Ctrl` + `S` to save the changes.

Related Information

[Repository Type Properties \[page 1012\]](#)

[Context \(Repository Type\) \[page 1021\]](#)

[Member Event Handling \[page 81\]](#)

[Provisioning \[page 98\]](#)

3.1.8.7 Configuring Contexts for a Repository Type

Context

Use the [Context](#) tab in the repository type properties to configure the context type for the repository type. This will be default configuration for all privileges which reference repositories of this repository type. It is possible to override the context configuration for specific privileges, if necessary.

Note

It is not recommended to change the context type once assigned. There is no automatic reconciliation process which will detect this, as the context types are assumed to be unchanged.

To define contexts for a repository type:

Procedure

1. Select the repository type in the tree view and view the properties by choosing *Open* from the context menu.
2. Select the *Context* tab.
3. Select the autoassign context policy and allowed context type(s) for the repository type.
4. Choose  *File*  or press `Ctrl` + `S` to save the changes.

Related Information

[Repository Type Properties \[page 1012\]](#)

[Context \(Repository Type\) \[page 1021\]](#)

[Context Based Assignments \[page 60\]](#)

3.1.8.8 Deleting a Repository Type

Context

If a repository type is no longer in use, you can delete it.

Note

When deleting the repository type, you have to decide what to do with any repositories of this type.

Procedure

1. Select the repository type in the tree view.
2. Choose *Delete* from the context menu.

The system checks for repositories of this type.

- If no repositories are found, the repository type is deleted.
- If repositories of this type are found, the [Confirm Delete](#) dialog box is displayed.
 - a. The list in the [Confirm Delete](#) shows the number of repositories of this type.
 - b. Decide what to do with repositories of this type. You can either keep them as orphaned repositories, or delete them.
 - If you decide to delete the repositories, all repositories of this type are removed from the system. For instance, jobs referencing these repositories will fail.
 - If you decide to keep the repositories, they will be in the system, but some constants defined on the repository type will not be present, meaning that any job or other reference to these constants will fail. If at a later point in time you add the repository type with the same name by importing another package or adding the repository type, the orphaned repositories will be reconnected to the repository type again, but the repository type constants will come from the new repository type.
 - c. Choose [OK](#).

Results

The repository type is removed from the system.

3.1.9 Maintaining Processes

A process defines a set of operations executed on entries stored in the identity store. A process can be public, meaning that it can be referenced and called from other packages.

Once a process has been created, it is visible in the tree view in the Identity Management Developer Studio. The content of a process is defined by tasks or other processes, which are executed in sequence. If any of them fails, no further tasks or processes are executed, and the process flow stops.

A number of different task types exist. Tasks and processes can be added to a process using a process flow diagram - a graphical editor making it easier to visualize the structure and the process flow.

Every process has a process type that is specified when the process is created. The process type defines the usage of the process in different events, such as validation, execution, notification or result handling. For example, Validate Add process, Notify process, Result Handling OK process and others.

The default value of the process type is [Regular](#). Regular processes are generic (unspecified) processes that can also be used as specific type of processes. For example, when you define an event process (Validate) on a repository type, role or privilege, instead of a Validate Add process, you can also use a Regular process.

Related Information

[General \(Process\) \[page 1086\]](#)

3.1.9.1 Creating a Process Folder

You can create process folders to better structure your processes in the configuration.

Context

To add a process folder:

Procedure

1. Select the process folder where you want to add the sub-folder, and choose ► **New** ► **Folder** ► from the context menu.

The new process folder is added in the tree view.

2. Enter a name for the process folder in the tree view.

Related Information

[Creating a Process \[page 220\]](#)

3.1.9.2 Creating a Process

Processes define a set of operations executed on entries that are stored in the identity store. The content of the processes is defined by tasks or other processes. A process can be public, meaning that it can be referenced and called from other packages.

Context

To create a process, proceed as follows:

Procedure

1. In the tree view in Identity Management Developer Studio, select the *Processes* folder from a package in your identity store.

Note

The package must be checked out and ready for editing.

2. Choose  *New*  *Process*  from the context menu.

The process is created and added in the tree view.

The properties of the process, the process flow diagram, and the tree view for the process structure are automatically opened and displayed once the process is created. You can start configuring the process.

3. You can modify the name of the process in the tree view. Select the process in the tree view and choose *Rename* from the context menu.

Related Information

[Viewing and Configuring Process Properties \[page 221\]](#)

[Managing Process Contents \[page 225\]](#)

3.1.9.3 Viewing and Configuring Process Properties

Context

To view or configure the process properties, perform the following steps:

Procedure

1. Select the process that you want to view or configure the properties of in the tree view in Identity Management Developer Studio and choose *Properties* from the context menu.

Note

If you have just created a process, its properties are opened automatically during creation.

2. Select a tab to enter or view the information about the process. The processing properties contain the following tabs:
 - *General*
 - *Result Handling*
 - *Documentation*

For more information and details about the tabs, and the properties and fields available, see *Process Properties*.

Note

The package must be checked out and ready for editing before you start changing the properties.

3. Choose ► *File* ► *Save* or press **Ctrl** + **S** to save the changes in the process properties.

Related Information

[Process Properties \[page 1086\]](#)

[Creating a Process \[page 220\]](#)

3.1.9.4 Viewing and Configuring the General Properties for a Process

Use the [General](#) tab in the process properties to enable/disable the process, define the process type or a repository for the process.

Context

To view or configure the general information of the process, proceed as follows:

Procedure

1. Select the process in the tree view.
2. Choose [Properties](#) from the context menu to open the process properties.
3. Select the [General](#) tab.
4. View the general information about the process or make changes to the properties. Enable or disable the process, define the process type or a repository, or define the use of context variables for the process. For more information about the fields and their options, see *General (Process)*.
5. Choose  [File](#)  or press Ctrl + S to save the changes in the process properties.

Related Information

[General \(Process\)](#) [page 1086]

3.1.9.5 Viewing and Configuring the Result Handling for a Process

To view or configure result handling for a process, you use the [Result Handling](#) tab in the process properties.

Context

To view or configure the result handling for a process, proceed as follows:

Procedure

1. Select the process in the tree view and choose [Properties](#) from the context menu.
2. Select the [Result Handling](#) tab.
3. View the result handling configuration or make any necessary changes.

Note

You can only make changes in the configuration if the package that the process belongs to is checked out and ready for editing.

For details about the fields on the [Result Handling](#) tab, see *Result Handling (Process)*.

4. Choose  [File](#)  or press Ctrl + S to save any changes that you have made.

Related Information

[Result Handling \(Process\) \[page 1089\]](#)

3.1.9.6 Viewing and Entering the Description of a Process

You can enter a description to be used as internal documentation of a process.

Context

To view or enter a process description, perform the following steps:

Procedure

1. Select the process in the tree view and choose *Properties* from the context menu.
2. Select the *Documentation* tab.
3. View or enter the process documentation.

Note

Before you can make changes in the configuration, the package that the process belongs to must be checked out and ready for editing.

4. Choose ► *File* ► *Save* or press `Ctrl` + `S` to save any changes you have made.

Related Information

[Documentation \(Task/Process\) \[page 1112\]](#)

3.1.9.7 Making a Process Public

If a process is public, it can be referenced and called from other packages.

Context

To make a process public, proceed as follows:

Procedure

1. Select the process in the tree view.
2. Choose *Make Public* from the context menu. Choose *OK* to close the information message.

Once the process has been made public, the qualified name for the package is added (see the *Qualified Name* field on the *General* tab of the process properties. If the process properties are already open when the process is made public, you might need to close and re-open the properties in order to update the *Qualified Name* field.

Note

A public process cannot be renamed. You need to remove it from public before you can rename it.

To remove a process from public, select it in the tree view and choose *Remove from Public* from the context menu. Choose *OK* to confirm. The *Qualified Name* field of the process should then be empty.

Related Information

[General \(Process\) \[page 1086\]](#)

3.1.9.8 Managing Process Contents

The contents of a process are defined by tasks or other processes. In other words, you can add other processes to a process or create tasks and add them as part of a process in Identity Management Developer Studio.

Prerequisites

Your package is checked out and ready for editing before making changes in the configuration.

Context

A number of different tasks exist. To add a task or a sub-process to a process, proceed as follows:

Procedure

1. Select the process in a package of your identity store where you want to create a task or add a sub-process in the tree view.
2. Double-click the selected process to open the flow diagram.

Note

The process flow diagram opens automatically when a process is created.

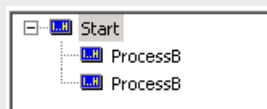
3. Use the process flow diagram to add the tasks or other existing processes to the process flow.

Note

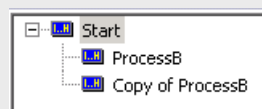
If one and the same sub-process is added several times (for example, through other processes) in a process flow, the sub-processes will not be executed because this results in unique constraint violation. As a workaround, you can copy the original sub-process as many times as you need and add the copies of the original sub-process in the process flow. Thus, you will add sub-processes with different task ID and task name.

See the example below, where *ProcessA* is the parent process and *ProcessB* is the sub-process:

Sub-processes will not be executed



Sub-processes will be executed



Note

If you have defined one process flow with switch or conditional tasks containing one and the same tasks in different branches, SAP Identity Management executes these tasks in different order than the defined one. To solve this, see SAP Note: [2973162](#) Tasks within Switch or Conditional tasks are executed in wrong order

For details on how to add or delete tasks and processes using the process flow diagram, see *Adding and Deleting Tasks in Process Flow Diagram* and *Adding and Deleting a Process in Process Flow Diagram*.

Note

Copy and paste of process tasks using the key combinations `Ctrl + C` and `Ctrl + V` is not supported. The result of its usage may not be the intended and other issues with the copied tasks may occur.

The following tasks can be added to a process flow:

Action Task	An action task is a task that references a job run when the task is executed. Normally, there will only be one pass in the job, a To-pass. The source of the pass will be an entry in the identity store that is ready to be processed by this task. The destination of the pass will be the repository that is referenced in the task properties.
Wait Task	A wait task gives you a possibility to influence the task execution flow. There are no properties to configure for the wait task, as it always waits for all tasks before it to complete before the next task is executed.
Switch Task	A switch task controls task execution by evaluating the result of a condition and executing different task(s) depending on the result. The task has a number of case nodes that correspond to the possible values of the returned result. Add as many case nodes as needed to cover the different values returned from the condition. The result of the query or stored procedure is compared to the names of the case nodes. If a matching node is found, the tasks below this node are executed in the order they are defined below the node. If no node matches the result of the condition, the tasks under the <i>Else</i> node are executed, if such a node is defined.
Conditional Task	The conditional task has a <i>True</i> and a <i>False</i> node. The tasks under the <i>True</i> node are executed if the result of the test specified in the condition is true. The tasks under the <i>False</i> node are executed if the result of the test specified in the condition is false.
Approval Task	The approval task has three sub-nodes, <i>Approve</i>, <i>Decline</i> and <i>Timeout</i>. The tasks under the <i>Approve</i> node are executed when the request is approved. The tasks under the <i>Decline</i> node are executed if the request is declined. The tasks under the <i>Timeout</i> node are executed if a timeout occurs. For assignment approvals, the status of the approval is processed according to the value of the MX_ATTR_STATE attribute of the pending value object when the approval task completes (the whole task tree has been executed). You can use the notification process to notify the involved parties in the approval as the approval is processed. The process referenced in the field <i>Execute Process on Initialize</i> can be used to send an e-mail to the approvers. There is a job template that can be used to create the action task/job.
Attestation Task	Attestation tasks are used to perform periodic attestations of roles (privilege) assignments. You can use the notification process to send the notifications to the attester. There is a job template that can be used to create the action task/job.

4. Save the process and the tasks/processes added to it by choosing ► *File* ► *Save* ► or pressing **Ctrl** + **S**.
5. To view or configure the properties of an added task, see *Viewing and Configuring Task Properties*.

Note

The wait task has no configurable properties.

Related Information

[Adding and Deleting Tasks in Process Flow Diagram \[page 231\]](#)

[Adding and Deleting a Process in Process Flow Diagram \[page 233\]](#)

[Viewing and Configuring Task Properties \[page 238\]](#)

[Creating a Process \[page 220\]](#)

3.1.9.8.1 About the Process Flow Diagram

Identity Management Developer Studio provides graphical process editing. This is done using the process flow diagram.

In other words, the process flow diagram enables you to graphically model the processes and their contents. The diagram opens in a graphical editor, and it is displayed as one of the windows in the editor (the [Process Flow Diagram](#) tab in the lower left corner). The other one being the tree view presentation of the graphical process flow diagram (the [Tree View](#) tab in the lower left corner, next to [Process Flow Diagram](#) tab). While the process flow diagram is actively used to model the process flow, the tree view is a read only tree view presentation which can be useful to get an overview.

The process flow diagram consists of two main parts (areas):

- a canvas background
- a palette menu

The Canvas

The canvas area is where the process flow is created and edited. It has a white, paper-like background color and a grid to help users align the added objects comfortably. The grid's colors are very subtle, so as not to distract attention from the flow objects like tasks and the connecting lines.

The Palette

The palette is a user interface control used to host reusable elements, which in this case are different types of tasks to be added as a part of the process flow (the palette contains a list of all available tasks that can be added). Editing is performed by adding the tasks from the palette onto the canvas, and setting the relationships among those tasks.

In Identity Management Developer Studio, the palette is positioned to the right of the canvas by default. If you have a different preference however, for example palette to the left, then this can be changed by right-clicking the [Palette](#) header and selecting ► [Dock On](#) ► [Left](#) ▾ from the context menu.

Flow Objects

As previously mentioned, the palette contains the list of tasks that are added to the process flow by adding them to the canvas. Tasks are one of the flow objects types you will work with in the process flow diagram. The flow object types are:

- **Tasks**
There is a number of tasks available: action task, wait task, switch task, conditional task, approval task and attestation task. On hover, a tool tip displays more information about the task, like name or GUID. In addition, a context button pad appears with options to delete the hovered task or add a new connection. Tasks that are newly added to the process flow, before the changes are saved, are displayed with the dotted red border. Once saved, they get the solid grey border.
- **Start/End node**
When viewing a process in the process flow diagram, you will always see a start node and an end node for the process. These nodes can never be deleted for the process. The tasks and their relationships are added between these start and end nodes.
Some of the tasks also have start and end nodes - task start and end nodes. The tasks that have a start and an end node are switch, conditional and approval tasks. These task start and end nodes, and all objects between them, are deleted along with the task when it is deleted.
- **Processes**
Other processes can be referenced as contents of a process. The process can be added in a process flow diagram by simple drag and drop, and then moved into the right position in the flow. A warning will be displayed as long as the process is not connected to the flow. Processes have a solid grey border which is wider than for the rest of the objects. On hover, XX. The referenced process will not be expanded in the flow it is added to, instead the details are displayed in a separate graphical editor window. Deleting a process from the flow will only remove the reference to the process in the flow diagram, and not actually delete the process from the database.
- **Connections**
Connections define sequence flow relationships between the tasks, and also within the tasks. All connections look the same, and have the same combination of line and arrow style. When a connection is selected it becomes dotted and yellow, and its connecting and bending points become visible. On hover, the connections don't change their appearance, but a tool tip displays more details about a connection, like for example a connection type. If, however, a task is selected when a connection is hovered, the connection becomes yellow and dotted but no connecting and bending points are shown.
The following types of connections are available:
 - **Order**
The order connection defines relationships between the tasks.
 - **Case/else**
Case connection and else connection are used within the switch task. A switch task can have a number of defined cases, and the case connections define relationships to task(s) executed for each of the cases. Else connection defines relationship to task(s) executed when none of the other defined cases match.
 - **True/false**
True and false connections are used within the conditional task. True connection defines relationship to task(s) executed when the condition for the conditional task is true. False connection defines relationship to task(s) executed when the condition for the conditional task is false.
 - **Approve/decline/timeout**
These types of connections are used within the approval task. Approve connection defines relationship to task(s) executed on approval. Decline connection defines relationship to task(s) executed on

decline. While the timeout connection defines relationship to task(s) executed when the approval times out.

When adding new objects to the process flow, you may either move the objects around on the canvas manually by selecting and dragging each individually, or use the [Layout Diagram](#) option from the context menu. There is a global setting to automatically layout the diagram. For more information about this, see [Configuring Automatic Layout of the Process Flow Diagram](#).

Related Information

[Configuring Automatic Layout of the Process Flow Diagram \[page 230\]](#)

[Adding and Deleting Tasks in Process Flow Diagram \[page 231\]](#)

3.1.9.8.2 Configuring Automatic Layout of the Process Flow Diagram

Each time a new object is added to the canvas, you would want to re-organize the process flow diagram. You may either do this manually, or define a global setting that does this automatically for you.

Context

The global setting [Auto Layout Process Flow Diagrams](#) needs to be activated. To do so, complete the following steps:

Procedure

1. In the Identity Management Developer Studio, choose **Window** > **Preferences**.

This will open the [Preferences](#) dialog box.

2. From the menu on the left, expand [SAP IdM](#) and select [Global Settings](#).
3. Select the global setting [Auto Layout Process Flow Diagrams](#).
4. Choose [Apply](#), and then [OK](#) to close the dialog box.

Automatic layout of process flow diagram will be applied each time you add a new object to the canvas in the diagram.

Related Information

[About the Process Flow Diagram \[page 228\]](#)

3.1.9.8.3 Adding and Deleting Tasks in Process Flow Diagram

Use the process flow diagram to graphically model the processes and their contents.

Context

To add tasks to and structure them within the process flow, do the following:

Procedure

1. Select one of the tasks from the palette to the right of the canvas.

Moving the cursor across the canvas you will now see a symbol which means that the selected task is attached.

2. Move the cursor to the connection where you want to add the task. When the connection is highlighted (yellow and dotted), click on it. The task will be automatically added and visible in the flow diagram. You may create and add as many tasks as you need into a process.

To delete a task, select it from the canvas and choose *Delete* from the context menu. Alternatively, choose the delete button (a trash can icon) from the context button pad when the task is hovered. You will be asked for a confirmation before the task is deleted.

3. Choose *Layout Diagram* from the context menu to layout the diagram for you, or move the task(s) and the connections manually where you want by selecting and dragging each individually. Optionally, you can use a global setting to automatically layout the diagram. For more information about this, see *Configuring Automatic Layout of the Process Flow Diagram*.
4. Save the process flow diagram by choosing **File > Save** or pressing **Ctrl** + **S**.

Related Information

[Managing Process Contents \[page 225\]](#)

[Configuring Automatic Layout of the Process Flow Diagram \[page 230\]](#)

3.1.9.8.4 Copying Action Tasks

You can copy an action task from one process to another within the same or other package or identity store.

Context

To copy an action task, proceed as follows:

Procedure

1. In the *Process Flow Diagram* view, select the action task and choose *Copy Action Task* from the context menu.
2. Check out the package where you want to paste the action task.
3. Open the target process and choose *Paste Action Task* from the context menu.

The action task appears in the upper left corner of the canvas. You can drag and drop it on the appropriate position in the process flow diagram.

Note

To preserve the pasted action task, save the *Process Flow Diagram*. If the copied action task is deleted in the meantime, the save operation will fail.

4. Choose *Save*.

Note

You cannot copy an action task to another SAP Identity Management database connection.

Note

Copy and paste of process tasks using the key combinations `Ctrl + C` and `Ctrl + V` is not supported. The result of its usage may not be the intended and other issues with the copied tasks may occur.

Results

Copied action tasks have new **Task IDs**, **Job IDs**, and **job status: <Idle>**. The rest of the copied action task properties are the same as its source. The job scripts are copied and all public package scripts are automatically relinked. The private package scripts are relinked automatically if they are located or referenced in the same package. Otherwise they are displayed with a warning sign in the *Error Status* column of the *Scripts* tab and you need to relink them manually.

Related Information

[SAP Note 3083807 Copying and Pasting Action Tasks in IDM is now possible](#) 

3.1.9.8.5 Adding and Deleting a Process in Process Flow Diagram

Use the process flow diagram to graphically model the processes and their contents.

Context

An already existing process in the configuration can be reused and added into a process flow of another process. In the process flow diagram, you add a process in the following way:

Procedure

1. Make sure to have the process flow diagram of the process you want to edit open. From the tree on the left side, navigate to and select a process that you would like to add to the process flow (under [Processes](#) in your package).
2. Drag the selected process and drop it onto the canvas.

The process will be visible on the canvas, with a warning that it has to be connected to the rest of the flow. The process is distinguished from other objects on the canvas by its border, which is solid grey and wider than the borders for other objects.

3. Select the process and connect it to the rest of the flow by simple drag and drop. When the connection where you want to add the process is highlighted (yellow and dotted), you can drop the process and it will be added.

Details of the added process will be displayed in a new graphical editor window. From the canvas, double-click the process on the canvas that you want to see the details of. You may select either [Process Flow Diagram](#) or [Tree View](#) to see the details.

Deleting a process from the process flow diagram will not actually delete the process itself, but only delete the reference to it in the flow. The process will still be available in the tree view on the left side under [Processes](#) in your package. To delete the process from the flow, select the process and choose [Delete](#) from the context menu. You will be asked for a confirmation before the process is deleted from the flow.

4. Choose [Layout Diagram](#) from the context menu to layout the diagram for you, or move the task(s) and the connections manually where you want by selecting and dragging each individually. Optionally, you can use a global setting to automatically layout the diagram. For more information about this, see [Configuring Automatic Layout of the Process Flow Diagram](#).
5. Save the process flow diagram by choosing  [File](#)  [Save](#) or pressing **Ctrl** + **S**.

Related Information

[Managing Process Contents \[page 225\]](#)

[Configuring Automatic Layout of the Process Flow Diagram \[page 230\]](#)

[About the Process Flow Diagram \[page 228\]](#)

3.1.9.8.6 Exporting the Process Flow Diagram as a Picture

You can export a process flow diagram as a picture in the following formats: BMP, GIF, JPG, PNG, RLE.

Context

To export a process flow diagram, proceed as follows:

Procedure

1. Open the process flow diagram on the canvas.
2. Select a task in the process flow diagram and choose [Export Diagram](#) from the context menu.
3. In the [Export Diagram](#) dialog box, you can do the following:
 - To export the whole process flow diagram, choose [All](#), select the image format and configure the size of the picture.
 - To export the single task you have selected, choose [Selected \(only single-selection\)](#), select the image format and configure the size of the picture.
4. Choose [OK](#).
5. In the [Save As](#) window, enter a file name and choose [Save](#).

→ Tip

Another way to export a whole diagram is to click on the canvas, choose [Export Diagram](#) from the context menu and follow the procedure above.

3.1.9.8.7 Viewing Process Contents

Context

You can use the [Tree View](#) tab next to the [Process Flow Diagram](#) tab to view the contents of a process. The contents of a process are defined by tasks or other processes.

Procedure

1. Select the process in a package of your identity store.
2. Double-click the selected process to open the flow diagram.
3. Select the [Tree View](#) tab next to the [Process Flow Diagram](#) tab.

The [Tree View](#) displays the content of the process.

4. You can do the following:
 - To expand a process, select it and choose [Expand All](#) from the context menu.
 - To view the properties of a process or a task, select it and choose [Properties](#) from the context menu.

Note

In the [Tree View](#) you cannot modify the content of a process. For example, you cannot rename a process or a task.

3.1.9.9 Testing a Process

When defining a process, you can test it along with its content in the Identity Management Developer Studio.

Prerequisites

- The logged-in user in the Identity Management Developer Studio must be a developer administrator or the owner of the package containing the process.
- The package does not need to be checked out.
- The dispatcher must be running.

Context

You can test a process and its content (tasks and child processes) by providing the MSKEYVALUE of an existing entry in the identity store. For example, you can test a provisioning process to update user attributes in a target system. Once you provide the MSKEYVALUE of a given user and start the test, SAP Identity Management will run the jobs referenced from the process for this given user. You can then view the logs for information about the execution of the process.

Procedure

1. Select a process and choose [Test](#) from the context menu.
2. Enter the MSKEYVALUE of an existing entry that you want to test the process for and choose [Test](#).
If the entry does not exist, you will get an error: `Could not find entry with MSKEYVALUE:<...>` and you cannot proceed with the test.
3. An [AuditID](#) is set for the current operation.
4. You can follow the processing of the process and its content. The columns contain the following information:

Column	Description
Step	The process ID and the name of the process.
State	The state of the process, for example: Ready to run, Expanded OK, Task Completed OK.
Retry Count	Number of retry attempts if an action task fails the first time. It is configured on the General tab of the task.
Update Time	The date and time the entry is written in the log.
Messages	Additional information about the processing of the process, for example: error messages or the result of tests for conditional tasks.

3.1.9.10 Copying a Process

You can copy a process from one package to another within the same identity store. To copy a process from one identity store to another, you need to export the package containing that process and import it to the identity store you want.

Prerequisites

- The package that you want to copy the process to is checked out.

Context

To copy a process within the same identity store, proceed as follows:

Procedure

1. In the tree view, select the process and choose [Copy](#) from the context menu.
2. Select the package that you want to copy the process to.
3. Select the [Processes](#) folder and choose [Paste](#) from the context menu.
4. Choose [Save](#).

Note

If the process that you want to copy to another package is referenced from another process, make sure this process is made public. If not, copy to will not be allowed. You will get a [Move Blocked](#) warning that the referenced object is not public.

3.1.9.11 Deleting a Process

You can delete processes that you do not need any more.

Context

To delete a process, proceed as follows:

Procedure

1. In the tree view in Identity Management Developer Studio, select and expand the [Processes](#) folder from a package in your identity store.

Note

Make sure that the package is checked out and ready for editing.

2. Select a process that you want to delete and choose [Delete](#) from the context menu.
3. A dialog box appears requesting confirmation. Choose [Yes](#) to confirm (or [No](#) to cancel) deletion of the selected process.
4. If the process you are trying to delete is referenced by other places in the configuration, a new deletion confirmation dialog box will appear, listing all the dependencies that will be affected by deletion of the process in question. The list displays the ID and name of the object that the selected process has references to, the type of reference, the current value, and what the new value will be if the process is deleted. Choose [OK](#) to confirm deletion of the process thus indicating that you accept the consequences, or choose [Cancel](#) to cancel deletion.

If you cancel deletion, the reference to the listed object will keep the value defined in the [Current Value](#) column.

If you proceed with deletion, the reference value will then be as specified in the [New Value](#) column.

3.1.9.12 Maintaining Task Properties

3.1.9.12.1 Viewing and Configuring Task Properties

Prerequisites

- Make sure that the package is checked out before editing.

- The process, containing the task which you want to view or configure the properties of, is opened in the task flow diagram.

Context

To view or configure the task properties, do the following:

Procedure

1. Select the task which you want to view or configure the properties of in the task flow diagram.
2. Choose *Open Properties* from the context menu for the selected task (or double-click the task).
3. Select a tab to enter or view the information about the task in the editor. What tabs are available depends on what task is selected.
4. Choose  *File*  or press `Ctrl` + `S` to save the changes in the task properties.

Related Information

[Task Properties \[page 1109\]](#)

3.1.9.12.1.1 Viewing and Configuring Action Task Properties

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the process flow diagram.

Context

To view or configure the action task properties, do the following:

Procedure

1. Select the action task which you want to view or configure the properties of in the process flow diagram.
2. Choose *Open Properties* from the context menu for the selected task (or double-click the task).
3. Select a tab to enter or view the information about the task in the editor.
 - *General*
 - *Result Handling*
 - *Logging*
 - *State*
 - *Documentation*
4. Choose  *File*  or press `Ctrl` + `S` to save the changes in the action task properties.

Related Information

[Action Task \[page 1129\]](#)

3.1.9.12.1.2 Viewing and Configuring Switch Task Properties

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the task flow diagram.

Context

To view or configure the switch task properties, do the following:

Procedure

1. Select the switch task which you want to view or configure the properties of in the task flow diagram.
2. Choose *Open Properties* from the context menu for the selected task (or double-click the task).
3. Select a tab to enter or view the information about the task in the editor.
 - *General*
 - *Result Handling*
 - *Documentation*
4. Choose ► *File* ► *Save* or press **Ctrl** + **S** to save the changes in the switch task properties.

Related Information

[Switch Task \[page 1115\]](#)

3.1.9.12.1.3 Viewing and Configuring Conditional Task Properties

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the task flow diagram.

Context

To view or configure the conditional task properties, do the following:

Procedure

1. Select the conditional task which you want to view or configure the properties of in the task flow diagram.
2. Choose *Open Properties* from the context menu for the selected task (or double-click the task).
3. Select a tab to enter or view the information about the task in the editor.
 - *General*
 - *Result Handling*
 - *Documentation*
4. Choose ► *File* ► *Save* or press **Ctrl** + **S** to save the changes in the conditional task properties.

Related Information

[Conditional Task \[page 1112\]](#)

3.1.9.12.1.4 Viewing and Configuring Approval Task Properties

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the task flow diagram.

Context

To view or configure the approval task properties, do the following:

Procedure

1. Select the approval task which you want to view or configure the properties of in the task flow diagram.
2. Choose *Open Properties* from the context menu for the selected task (or double-click the task).
3. Select a tab to enter or view the information about the task in the editor.
 - *General*
 - *Result Handling*
 - *Attributes*
 - *Approval*
 - *Presentation*
 - *Documentation*
4. Choose ► *File* ► *Save* or press **Ctrl** + **S** to save the changes in the approval task properties.

Related Information

[Approval Task \[page 1117\]](#)

3.1.9.12.1.5 Viewing and Configuring Attestation Task Properties

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the task flow diagram.

Context

To view or configure the attestation task properties, do the following:

Procedure

1. Select the attestation task which you want to view or configure the properties of in the task flow diagram.
2. Choose [Open Properties](#) from the context menu for the selected task (or double-click the task).
3. Select a tab to enter or view the information about the task in the editor.
 - [General](#)
 - [Result Handling](#)
 - [Attestation](#)
 - [Documentation](#)
4. Choose  [File](#)  or press `Ctrl` + `S` to save the changes in the attestation task properties.

Related Information

[Attestation Task \[page 1126\]](#)

3.1.9.12.2 Specifying the General Parameters of an Action Task

To configure the general parameters of an action task, you use the [General](#) tab of the action task properties.

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the process flow diagram.

Context

To configure the general parameters, do the following:

Procedure

1. Select the action task you want to configure in the process flow diagram and open the properties. Choose [Open Properties](#) from the context menu for the action task or double-click the task.
2. Select the [General](#) tab to enter the information about the action task. You may:

Option	Description
Enable or disable the task	Enabling or Disabling a Task [page 245]
Specify the use of parallel provisioning	Select the option Allow Parallel Provisioning to activate the use of parallel provisioning. Otherwise keep deselected.
Specify dispatcher(s)	Select dispatcher(s) in the field Run by Dispatcher(s) .
Specify retries for the task	Define the fields No. of Retries and Retry Delay .

3. Choose [File](#) > [Save](#) or press `Ctrl` + `S` to save the changes. Choose the [Advanced](#) button to open the [Advanced](#) tab and specify the advanced options for the runtime engine.

Related Information

[General \(Action Task\) \[page 1129\]](#)

3.1.9.12.3 Enabling or Disabling a Task

You can enable or disable a task by selecting the respective option from the right-click menu or by using the [General](#) tab of the task properties.

Prerequisites

- Make sure that the package is checked out before editing.

Context

There are two options to enable or disable a task. To do so, choose an approach from below:

- Using the right-click menu
Right-click on the task and select the *Enable* or *Disable* option from the menu.
- Using the *General* tab of the task properties
 - Select the task you want to enable/disable in the task flow diagram and open the properties. To do so, choose *Open Properties* from the context menu for the task, or just double-click the task.
 - Select the *General* tab.
 - Select the option *Enabled* to enable the task, or deselect the same option to disable the task.
 - Choose **File** > **Save** or press **Ctrl** + **S** to save the changes.

The enabled and disabled tasks have different visualization.

In the *Process Flow Diagram*, the tasks are displayed as follows:

- The enabled tasks are displayed in pale blue white gradient color with solid grey border.
- The disabled tasks are displayed in white color with thin grey border.

In the *Tree View*, the task name of the disabled task is followed by '(Disabled)' suffix.

Note

When a switch task, a conditional task or an approval task is enabled or disabled, the visualization of its nested tasks is not changed.

Related Information

[General \(Action Task\) \[page 1129\]](#)

[General \(Attestation Task\) \[page 1126\]](#)

[General \(Approval Task\) \[page 1118\]](#)

[General \(Switch Task\) \[page 1115\]](#)

[General \(Conditional Task\) \[page 1113\]](#)

[Specifying the General Parameters of an Action Task \[page 244\]](#)

3.1.9.12.4 Specifying a Repository for a Task

Select the *General* tab of the task properties to specify a repository for the task.

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the process flow diagram.

Context

Do the following:

Procedure

1. Select the task you want to specify a repository for in the process flow diagram and open the properties. Choose *Open Properties* from the context menu for the task, or just double-click the task.
2. Select the *General* tab.
3. Define the repository for the task in the *Repository* field. Choose the ... button, select an available repository from the list and choose *OK* to add it.
4. Choose **File** > *Save* or press **Ctrl** + **S** to save the changes.

Related Information

[General \(Attestation Task\) \[page 1126\]](#)

[General \(Approval Task\) \[page 1118\]](#)

[General \(Switch Task\) \[page 1115\]](#)

[General \(Conditional Task\) \[page 1113\]](#)

3.1.9.12.5 Configuring Result Handling for a Task

To configure result handling for a task, you use the [Result Handling](#) tab of the task properties.

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the task flow diagram.

Context

Based on the result of the task, it is possible to start tasks. Tasks can be called internally from other tasks, or externally, for instance based on an attribute definition in the identity store. There are two types of task result handling: task result actions and chain result actions. The rules for when the result handling is performed, depends on the type of result handling:

- Task result actions are executed based on the result of this task, regardless from where the task is called.
- A chain result action is only executed if the task is called externally, based on the result of all sub-tasks.

Note

Chain result action is not available for action tasks.

To configure the result handling, do the following:

Procedure

1. Select the task in the task flow diagram and open the properties. Choose [Open Properties](#) from the context menu for the task, or just double-click the task.
2. Select the [Result Handling](#) tab.
3. Configure the result handling for the task. You may:
 - Define a task that will be executed if this task completes successfully (task result action for OK result).
 - Define a task that will be executed if this task fails (task result action for failed result).
 - Define a task that will be executed if this task and all sub-tasks complete successfully (chain result action for OK result). This is not available for the action tasks.
 - Define a task that will be executed if this task or any sub-tasks fail (chain result action for failed result). This is not available for the action tasks.
4. To select and add tasks for task result action and chain result action, use the ... button to the right of the fields.

5. Choose ► **File** ► **Save** or press **Ctrl** + **S** to save the changes.

Related Information

[Result Handling \(Tasks\) \[page 1111\]](#)

3.1.9.12.6 Configuring Logging for an Action Task

Use the [Logging](#) tab to configure the logging for an action task.

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the task flow diagram.

Context

A set of default values for the log options are defined. If necessary, you can change these:

Procedure

1. Select the action task in the task flow diagram and open the properties. Choose [Open Properties](#) from the context menu for the task, or just double-click the task.
2. Select the [Logging](#) tab.
3. Change the log level or other options for the log.

For details about the logging options and fields, see the field descriptions in [Logging \(Job/Action Task\)](#).

4. Choose ► **File** ► **Save** or press **Ctrl** + **S** to save the changes.

Related Information

[Logging \(Job/Action Task\) \[page 1024\]](#)

[Action Task \[page 1129\]](#)

3.1.9.12.7 Viewing a State of an Action Task

Select the [State](#) tab of action task properties to view the state properties.

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the task flow diagram.

Context

Do the following:

Procedure

1. Select the action task in the task flow diagram and open the properties. Choose [Open Properties](#) from the context menu for the task, or just double-click the task.
2. Select the [State](#) tab.
3. View the state properties. You may have to choose the [Refresh](#) button to update some of the information.

For details about the state properties and the fields, see the field descriptions in *State (Job/Action Task)*.

Related Information

[State \(Job/Action Task\) \[page 1025\]](#)

[Action Task \[page 1129\]](#)

3.1.9.12.8 Specifying the Conditions for a Switch Task

A switch task controls the task flow by evaluating the result of a condition. You can specify the condition on the [General](#) tab of the switch task properties.

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the task flow diagram.

Context

The case node matching the result of the condition will be executed. If no node matches the result, the [Else](#) node will be executed, if it exists.

The conditions can be specified using:

- **Stored procedure**
The stored procedure receives [MSKEY](#), [Repository ID](#) and [Task ID](#) as parameters. It returns a string that is compared to the defined case nodes. The database user that is used by the runtime engine (normally [mxmc_rt](#)) must have access to the stored procedure.
- **Filter**
You can use a filter to perform a test and return a value that will be used to select the corresponding case node.
- **SQL query**
You can use an SQL query to perform a test and return a value that will be used to select the corresponding case node.

Procedure

1. Select the switch task in the task flow diagram and open the properties. Choose [Open Properties](#) from the context menu for the task, or just double-click the task.
2. Select the [General](#) tab.
3. Use either the stored procedure, filter or the SQL query to specify the condition:

Option	Description
Stored Procedure	Select this option, and then select a stored procedure in the list that appears. This list contains all stored procedures with a prefix mxpt_ defined in the Identity Center database.

Option	Description
Filter	Select the option and then enter the filter in the field that appears. You can use the filter editor to create the filter (the Edit... button just above the right upper corner of the field). The filter will automatically generate SQL query. To view the query, select the SQL Query option. For details on how to use the filter editor, see <i>Using the Filter Editor</i>. You can also choose to include inactive entries or use attribute IDs instead of attribute names (options Include Inactive Entries and Use Attribute ID Instead of Attribute Name in SQL Query respectively). In addition, you can specify the attribute to be returned (the Return Attribute field). Choose the ... button to the right of the field to browse for and select the attribute. The value is used to select the correct case node. Note If you select the SQL Query option and then manually edit the query, the Filter option will be disabled. To activate Filter again, you must clear the SQL query field.
SQL Query	Select the option and then enter the SQL in the field that appears. You may use the Filter option to help you generate an SQL query, that you so can extend. Note If you manually edit the SQL query, the Filter option will be disabled. To activate Filter again, you must clear the SQL query field. Note The query must use the Identity Management database, and the database user that is used by the runtime engine must have access to it. If the condition retrieves (reads) values from the identity store, it is recommended to use one of the views installed with the product. The query must be in the form: <pre>SELECT <column> FROM <view> WHERE <expression></pre> The SQL query must return a string that can be compared to the case nodes. Note When using Microsoft SQL Server, it is recommended that all SELECT queries should include "with (nolock)". This will reduce the table locking, improve performance, and reduce the probability of deadlocks. For instance: <pre>SELECT <columns> FROM <table> with (nolock) WHERE <column>='<value>'</pre> You can use the following variables in the query: • <code>%MSKEY%</code> The current MSKEY of the current entry. • <code>%REPID%</code> The current repository ID.

Option	Description
	• %TASKID% The current task ID. • %AUDITID% The audit ID of the current operation. • %NOLOCK% Checks if the database is Microsoft SQL Server and replaces the variable with "with (NO-LOCK)". If other database systems are used, the variable will be replaced with an empty string. For instance: <pre>SELECT avalue FROM idmv_value_basic %NOLOCK% WHERE attrname='MX_LOCATION' AND is_id = 1 AND mskey=%MSKEY%</pre>

4. Choose **File > Save** or press **Ctrl + S** to save the changes.

Related Information

[General \(Switch Task\) \[page 1115\]](#)

[Using the Filter Editor \[page 269\]](#)

3.1.9.12.9 Specifying the Conditions for a Conditional Task

A conditional task controls the task flow by evaluating the result of a test or a condition. You can specify the condition in the *General* tab of the conditional task properties.

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the task flow diagram.

Context

If the test returns values *0* or *""* (empty string), the *False* task branch is executed. If it returns any other value, the *True* task branch is executed.

The conditions can be specified using:

- **Stored procedure**
The stored procedure receives *MSKEY*, *Repository ID* and *Task ID* as parameters. It returns a string containing *0* (false) or *1* (true). The database user that is used by the runtime engine (normally *mxmc_rt*) must have access to the stored procedure. There is a sample procedure in the Identity Center database called *mxpt_test* that can be used as a template.
- **Filter**
You can use a filter to perform a test and return a true/false result.
- **SQL query**
You can use an SQL query to perform a test and return a true/false result.

Do the following:

Procedure

1. Select the conditional task in the task flow diagram and open the properties. Choose *Open Properties* from the context menu for the task, or just double-click the task.
2. Select the *General* tab.
3. Use either the stored procedure, the filter or the SQL query to specify the condition:

Option	Description
<i>Stored Procedure</i>	Select this option, and then select a stored procedure in the list that appears. This list contains all stored procedures with a prefix <i>mxpt_</i> defined in the Identity Center database.
<i>Filter</i>	Select the option and then enter the filter in the field that appears. You can use the filter editor to create the filter (the <i>Edit...</i> button just above the right upper corner of the field). The filter will automatically generate SQL query. To view the query, select the <i>SQL Query</i> option. For details on how to use the filter editor, see <i>Using the Filter Editor</i> . You can also choose to include inactive entries or use attribute IDs instead of attribute names (options <i>Include Inactive Entries</i> and <i>Use Attribute ID Instead of Attribute Name in SQL Query</i> respectively).
Note If you select the <i>SQL Query</i> option and then manually edit the query, the <i>Filter</i> option will be disabled. To activate <i>Filter</i> again, you must clear the SQL query field.	
<i>SQL Query</i>	Select the option and then enter the SQL in the field that appears. You may use the <i>Filter</i> option to help you generate an SQL query, that you so can extend.
Note If you manually edit the SQL query, the <i>Filter</i> option will be disabled. To activate <i>Filter</i> again, you must clear the SQL query field.	

Option	Description

Note

The query must use the Identity Management database, and the database user that is used by the runtime engine must have access to it.

If the condition retrieves (reads) values from the identity store, it is recommended to use one of the views installed with the product.

The query must be in the form:

```
SELECT COUNT(*) FROM <view> WHERE <expression>
```

The SQL query returns 0 (false) or a number greater than 0 (true).

Note

When using Microsoft SQL Server, it is recommended that all SELECT queries should include "with (no-lock)". This will reduce the table locking, improve performance, and reduce the probability of deadlocks. For instance:

```
SELECT <columns> FROM <table> with (nolock) WHERE <column>='<value>'
```

You can use the following variables in the query:

- **%MSKEY%**
The current MSKEY of the current entry.
- **%REPID%**
The current repository ID.
- **%TASKID%**
The current task ID.
- **%AUDITID%**
The audit ID of the current operation.
- **%NOLOCK%**
Checks if the database is Microsoft SQL Server and replaces the variable with "with (NOLOCK)". If other database systems are used, the variable will be replaced with an empty string.

For instance:

```
SELECT COUNT(*) FROM idmv_value_basic %NOLOCK% WHERE MSKEY=%MSKEY%  
AND attrname='MX_LOCATION' AND searchvalue like 'Trond%'
```

4. Choose **File > Save** or press **Ctrl + S** to save the changes.

Related Information

[General \(Conditional Task\) \[page 1113\]](#)

[Using the Filter Editor \[page 269\]](#)

3.1.9.12.10 Defining Attributes for the Identity Management User Interface for an Approval Task

When you configure approval task properties, you use the [Attributes](#) tab to configure which attributes are displayed for this task in the Identity Management User Interface.

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the task flow diagram.

Context

ⓘ Note

The [Attribute](#) tab is only available if you have selected the [Show Attributes](#) option on the [Approval](#) tab. The attributes you select here are displayed as part of the approval task to enable the approver to fill in additional fields when approving the request.

When specifying attributes for an approval task, it is not possible to do any assignments in the approval task, so assignment attributes cannot be included in the list.

Do the following:

Procedure

1. Select the approval task in the task flow diagram and open the properties. Choose [Open Properties](#) from the context menu for the task, or just double-click the task.
2. Select the [Attributes](#) tab.
3. Define the entry type for this task and select the attributes in the list. The list contains all attributes defined for the selected entry type.
 - Select [List](#) to specify that the attribute is to be included in the Identity Management User Interface.
 - Select [Mandatory](#) to specify that the attribute must be filled in before the task can be executed.
 - Select [Read Only](#) to specify that this attribute is read-only in the web interface.
 - Enter a default value for the attribute.

The default value will be used in the following way:

Default values are only used if the attribute does not already contain a value.

If the attribute is visible, the default value will be filled in when the task is displayed. If the attribute is invisible, the default value will be stored in the identity store without being displayed. This can be used to set system attributes or other attributes that should not be visible to the user.

You can also combine default values with read-only attributes. Then the default value is displayed in the web interface and stored in the identity store, but the user cannot modify the attribute.

4. For all listed attributes, you can change some of the properties of the attribute by choosing ... to the right of the selected attribute's name. This opens the *Identity Store Attribute* dialog box. Only the relevant properties are enabled.
5. Specify the order in which the attributes will be listed in the web interface by moving the attributes up and down in the list with the *Up* and *Down* buttons to the right of the list. You can also move the attributes up or down by dragging the attribute name with the mouse. The attributes are displayed in the same order as they have here.
6. Choose ► *File* ► *Save* ► or press **Ctrl** + **S** to save the changes.

Related Information

[Attributes \(Approval Task\) \[page 1119\]](#)

3.1.9.12.11 Configuring the Approval

When you configure an approval task to approve an assignment request, attribute modification, entry creation or other change that requires an approval, you use the *Approval* tab to configure the details about the approval.

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the approval task which you want to view or configure the properties of, is opened in the task flow diagram.

The process must be public.

You have set the correct process type of the process, containing the approval task. See [General \(Process\) \[page 1086\]](#).

❖ Example

Basic Approvals

To approve attribute modification, use *Modify Attribute Process* type. To approve entry creation, use *Add Entry Process* type.

Assignment Approvals

To approve whether a role or a privilege assignment is allowed for a given entry, use *Validate Add Process* type.

- Make sure that the [Evaluate Approvals](#) checkbox is selected on the [Policy](#) tab of the dispatcher.

Context

To configure the details about the approval, proceed as follows:

Procedure

1. Select the approval task in the task flow diagram and open the properties. Choose [Open Properties](#) from the context menu for the task, or just double-click the task.
2. Select the [Approval](#) tab.
3. Define the details for the approvals, like the approval type, a list of approvers, timeout, escalations and notifications. For more details about the options and fields that can be configured, see the field descriptions in [Approval \(Approval Task\) \[page 1119\]](#).
4. Choose  [File](#)  [Save](#) or press `Ctrl` + `S` to save the changes.

Note

Due to the specifics and the design of the Approval mechanism, we do not support the combination of grouping and approvals and we do not recommend using them together. However, if you decide to combine them, some faulty behavior may be observed. For more information, see [3196414](#) .

Next Steps

After defining the details for the approvals, add the process containing the approval task to your configuration wherever you need it to be executed.

Example

Basic Approvals

To approve attribute modification, add the [Modify Attribute Process](#) containing the approval task on the [Event Processes](#) tab of the specific attribute. To approve entry creation, add the [Add Entry Process](#) containing the approval task on the [Event Processes](#) tab of the specific entry type.

Assignment Approvals

To approve whether a role or a privilege assignment is allowed for a given entry, add the [Validate Add Process](#) containing the approval task on the [Member Events](#) tab of the specific role or privilege.

Related Information

[Approval Processing \[page 34\]](#)

3.1.9.12.11.1 Configuring the Approvers of an Assignment Approval

When you configure an approval, you define potential approvers. This is done on the [Approval](#) tab of the approval task properties.

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the flow diagram.

Context

How the approvers are defined depends on what approval type is selected. There are two types of approval: basic approval and assignment approval. This topic describes how to configure approvers of an assignment approval. For configuration of the approvers for a basic approval, see [Configuring the Approvers of a Basic Approval \[page 261\]](#).

Proceed as follows:

Procedure

1. Select the approval task in the flow diagram and open the properties. Choose [Open Properties](#) from the context menu for the task, or just double-click the task.
2. Select the [Approval](#) tab.
3. For an assignment approval (if the [Assignment](#) option is selected in the [Approval Type](#) field) you use the field [Get Approvers from](#) to define where to find the approvers of the approval request. Choose one of the following options:

Option	Description
Assignment Context	If you choose Assignment Context in the Get Approvers from field, the approvers (also escalation approvers) are found on the specific context that is part of the assignment.

Option	Description
	The approvers are specified with the attribute <code>MX_APPROVERS</code>. If you have chosen <i>Escalate to New List</i> in the <i>Escalation Rule</i> field on the task, you define the escalation approvers with the attributes <code>MX_ESCALATION_APPROVERS_1/2/3</code> (for each level). The number of escalation levels and timeout are defined on the task (the fields <i>Max No. of Escalations</i> and <i>Escalation Timeout</i>). You can add values to these attributes either with a job or by configuring a form that contains the attributes.
<i>Pending Value Object</i>	If you choose <i>Pending Value Object</i> in the <i>Get Approvers from</i> field, the approvers (also escalation approvers) are found on the pending value object that is created when the member event process is started. The approvers are specified with the attribute <code>MX_APPROVERS</code>. If there are any approvers defined on the role or privilege that is assigned, these are initially copied to the pending value object when it is created. The list of approvers can be modified by tasks executed before the approver task in the task structure of the member event process. If you have chosen <i>Escalate to New List</i> in the <i>Escalation Rule</i> field on the task, you define the escalation approvers with the attributes <code>MX_ESCALATION_APPROVERS_1/2/3</code> (for each level). The number of escalation levels and timeout are defined on the task (the fields <i>Max No. of Escalations</i> and <i>Escalation Timeout</i>). You can add values to these attributes by tasks executed before the approval task in the task structure of the member event process.
<i>Role/Privilege</i>	If you choose <i>Role/Privilege</i> in the <i>Get Approvers from</i> field, the approvers (also escalation approvers) are found on the role or privilege. When specifying approvers for a specific role or privilege, you use the attribute <code>MX_APPROVERS</code>. You can add values to this attribute either with a job or by configuring a form that contains this attribute. You could use the <i>Approvers</i> tab of the role or privilege properties. If you have chosen <i>Escalate to New List</i> in the <i>Escalation Rule</i> field on the task, you define the escalation approvers with the attributes <code>MX_ESCALATION_APPROVERS_1/2/3</code> (for each level). The number of escalation levels and timeout are defined on the task (the fields <i>Max No. of Escalations</i> and <i>Escalation Timeout</i>). You can add values to these attributes either with a job or by configuring a form that contains the attributes.
<i>Task</i>	If you choose <i>Task</i> as the source of the approvers in the <i>Get Approvers from</i> field, you use the <i>Approvers</i> field to define the approvers. To do so, enter the <code>MSKEYVALUE</code> of an entry in the identity store in the field. This could be a user, a role or a privilege. To verify the name of the entered approver, choose <i>Check Name...</i> to the right of the field. If only one entry is found, the value will be underlined. If more than one entry matches the value, a dialog box is displayed where you can select one of the entries. Only the 50 first entries are listed. If no entries match the criterion, a message is displayed and you must enter another value.

4. Choose  *File*  *Save* or press `Ctrl` + `S` to save the changes.

Related Information

[Approval Processing \[page 34\]](#)

[Approval \(Approval Task\) \[page 1119\]](#)

3.1.9.12.11.2 Configuring the Approvers of a Basic Approval

When you configure an approval, you define potential approvers. This is done on the [Approval](#) tab of the approval task properties.

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the flow diagram.

Context

How the approvers are defined depends on what approval type is selected. There are two types of approval: basic approval and assignment approval. This topic describes how to configure approvers of a basic approval. For configuration of the approvers for an assignment approval, see [Configuring the Approvers of an Assignment Approval \[page 259\]](#).

Proceed as follows:

Procedure

1. Select the approval task in the flow diagram and open the properties. Choose [Open Properties](#) from the context menu for the task, or just double-click the task.
2. Select the [Approval](#) tab.
3. For a basic approval (if the [Basic](#) option is selected in the [Approval Type](#) field) you define the approvers with approval rules on the task. Use the [Approvers](#) field - the list contains all approval rules that are defined for this task.
 - Define an approval rule by choosing [Add](#).
 - Modify an approval rule by selecting a rule in the list and choosing [Modify](#).
 - Remove an approval rule by selecting a rule in the list and choosing [Remove](#).

Note

For details about what information can be configured when adding or modifying an approval rule, see [Adding or Modifying an Approval Rule for Basic Approvals \[page 262\]](#).

4. Choose **File** > **Save** or press **Ctrl** + **S** to save the changes.

Related Information

[Approval Processing \[page 34\]](#)

[Approval \(Approval Task\) \[page 1119\]](#)

3.1.9.12.11.2.1 Adding or Modifying an Approval Rule for Basic Approvals

When you define basic approvals you specify how approval requests should be handled. To do this you add one or more approval rules. You must define rules so as to handle approval requests from all users.

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the flow diagram.
- You have opened the properties of the approval task in the flow diagram and are configuring the approvers for a basic approval on the [Approval](#) tab.

Context

To add or modify an approval rule, do the following steps:

Procedure

1. On the [Approval](#) tab, use the [Approvers](#) field to maintain the list of approval rules defined for this task.

To add an approval rule, choose [Add](#) to open the [Task Approval](#) dialog box.

2. Fill in the necessary information in the dialog box. The fields contain the following information:

Option	Description
Approver	Select the approver for this approval rule. You can choose between: • Logged-In User or Identity Store Entry One specific user/identity store entry is responsible for handling approval requests. • Referral Specify that the user referred to by a referral attribute is responsible for handling approval requests. This may typically be a manager/secretary or any other person/role responsible for handling approval requests for other users. • Filter Use a filter to specify who should be responsible for handling approval requests. Note This option is only available if you have defined filters in SAP Identity Management 7.2 and upgrade to version 8.0. For performance reasons we recommend that you do not use this option. You can use Logged-In User or Identity Store Entry and select a dynamic group instead. You use the fields below to specify the details of the approver. Depending on what you select in this list, the corresponding fields will be enabled.
ID Store	This list contains all available identity stores. If the approval rule should apply to a user or an entry from another identity store, select this identity store in the list. The contents of the user and entry lists change accordingly.
Name	This field is enabled if you select Logged-In User or Identity Store Entry in the Approver list. Select the user or the identity store entry who should handle approval requestes according to this approval rule. This user or the entry will see all approval requests for this task from the defined approvee(s) in the Identity Management User Interface. Enter MSKEYVALUE from the identity store. To verify the name, choose Check Name.... If only one entry is found, the value will be underlined. If more than one entry matches the value, a dialog box is displayed where you can select one of the entries. Only the 50 first entries are listed. If no entries match the criterion, a message is displayed and you must enter another value.
Referral Attribute	This field becomes visible when you select Referral in the Approver list. Select the referral attribute that is to be used to find the approver for the approvee(s) specified in this approval rule.
Filter	This field is enabled if you select Filter in the Approver list. Use the filter to create a query that can be used to identify the users who should be responsible for handling approval requests according to this approval rule. All users matching this query will see all approval requests for this task from the specified approvee(s) in the Identity Management User Interface. Select the option and then enter the filter in the field that appears. You can use the filter editor to create the filter (the Edit... button just above the right upper corner of the field). The filter will automatically generate SQL query. To view the query, select the SQL Query option. For details on how to use the filter editor, see <i>Using the Filter Editor</i>.

Option	Description
	You can also choose to include inactive entries or use attribute IDs instead of attribute names (options <i>Include Inactive Entries</i> and <i>Use Attribute ID Instead of Attribute Name in SQL Query</i> respectively). Note If you select the <i>SQL Query</i> option and then manually edit the query, the <i>Filter</i> option will be disabled. To activate <i>Filter</i> again, you must clear the SQL query field.
<i>SQL Query</i>	This field is enabled if you select <i>Filter</i> in the <i>Approver</i> list. Use this field to create an SQL query that can be used to identify the users who should be responsible for handling approval requests according to this approval rule. All users matching this query will see all approval requests for this task from the specified approvee(s) in the Identity Management User Interface. The SQL query must return the MSKEY of the user. Select the option and then enter the SQL in the field that appears. You may use the <i>Filter</i> option to help you generate an SQL query, that you so can extend. Note If you manually edit the SQL query, the <i>Filter</i> option will be disabled. To activate <i>Filter</i> again, you must clear the SQL query field. If you enter the query manually, it is recommended to use one of the views installed with the product. Note When using Microsoft SQL Server, it is recommended that all SELECT queries should include "with (nolock)". This will reduce the the table locking, improve performance, and reduce the probability of deadlocks. For instance: <pre>SELECT <columns> FROM <table> WITH (NOLOCK) WHERE <column>= '<value>'</pre>
<i>Approvee</i>	Select the users, from whom the approver should be responsible for approving requests. You can choose between the following values: <i>Everybody</i> Select this value to specify that the approver is responsible for handling approval requests from all users. <i>User of Group</i> The approver is responsible for approving requests from a specified user or group. <i>Filter</i> User a filter to specify who the approver should be responsible for approving the requests from. You can use the fields below to specify the details of the approvee. Depending on what you select in this list, the corresponding fields will be enabled.
<i>Name</i>	This field is enabled if you select <i>User or Group</i> in the <i>Approvee</i> list.

Option	Description
	Select the user or group from whom the approver should be responsible for handling approval requestes. Enter MSKEYVALUE from the identity store. To verify the name, choose Check Name... If only one entry is found, the value will be underlined. If more than one entry matches the value, a dialog box is displayed where you can select one of the entries. Only the 50 first entries are listed. If no entries match the criterion, a message is displayed and you must enter another value.
Filter	This field is enabled if you select Filter in the Approvee list. Use the filter to create a query that can be used to identify the users from whom the approver should be responsible for handling approval requests. Select the option and then enter the filter in the field that appears. You can use the filter editor to create the filter (the Edit... button just above the right upper corner of the field). The filter will automatically generate SQL query. To view the query, select the SQL Query option. For details on how to use the filter editor, see Using the Filter Editor. You can also choose to include inactive entries or use attribute IDs instead of attribute names (options Include Inactive Entries and Use Attribute ID Instead of Attribute Name in SQL Query respectively). Note If you select the SQL Query option and then manually edit the query, the Filter option will be disabled. To activate Filter again, you must clear the SQL query field.
SQL Query	This field is enabled if you select Filter in the Approvee list. Use this field to create an SQL query that can be used to identify the users from whom the approver should be responsible for handling approval requests. The SQL query must return the MSKEY of the user. Select the option and then enter the SQL in the field that appears. You may use the Filter option to help you generate an SQL query, that you so can extend. Note If you manually edit the SQL query, the Filter option will be disabled. To activate Filter again, you must clear the SQL query field. If you enter the query manually, it is recommended to use one of the views installed with the product. Note When using Microsoft SQL Server, it is recommended that all SELECT queries should include "with (nolock)". This will reduce the the table locking, improve performance, and reduce the probabllity of deadlocks. For instance: <pre>SELECT <columns> FROM <table> WITH (NOLOCK) WHERE <column>= '<value>'</pre>

- Choose [OK](#) to close the dialog box. This will add the specified information into the [Approvers](#) field.
- Choose [File](#) > [Save](#) or press Ctrl + S to save the changes.

Related Information

[Approval \(Approval Task\) \[page 1119\]](#)

[Configuring the Approvers of a Basic Approval \[page 261\]](#)

[Using the Filter Editor \[page 269\]](#)

3.1.9.12.12 Configuring the Presentation of the Approval Task in the Identity Management User Interface

When you configure approval task properties, you use the *Presentation* tab to configure how the task is presented in the Identity Management User Interface.

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the task flow diagram.

Context

Do the following:

Procedure

1. Select the approval task in the task flow diagram and open the properties. Choose *Open Properties* from the context menu for the task, or just double-click the task.
2. Select the *Presentation* tab.
3. Configure the presentation parameters, like the display name, task header and description, and buttons. For more details about the parameters that can be configured, see the field descriptions in *Presentation (Approval Task)*.
4. Choose  *File*  *Save* or press `Ctrl` + `S` to save the changes.

Related Information

[Presentation \(Approval Task\) \[page 1125\]](#)

3.1.9.12.13 Configuring the Attestation

When you configure an attestation task, you use the [Attestation](#) tab to configure the details about the attestation.

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the task flow diagram.

Context

Proceed as follows:

Procedure

1. Select the attestation task in the task flow diagram and open the properties. Choose [Open Properties](#) from the context menu for the task, or just double-click the task.
2. Select the [Attestation](#) tab.
3. Define the details for the attestation, like the list of attesters, timeout, notifications and reminders. For more details about the options and fields that can be configured, see the field descriptions in [Attestation \(Attestation Task\)](#).
4. Choose  [File](#)  [Save](#) or press `Ctrl` + `S` to save the changes.

Related Information

[Attestation \(Attestation Task\) \[page 1127\]](#)

3.1.9.12.14 Entering a Description of a Task

You can enter a description that is used as internal documentation of a task.

Prerequisites

- Make sure that the package is checked out before editing.
- The process, containing the task which you want to view or configure the properties of, is opened in the task flow diagram.

Context

To document the task:

Procedure

1. Select the task in the task flow diagram and open the properties. Choose *Open Properties* from the context menu for the task, or just double-click the task.
2. Select the *Documentation* tab.
3. Enter the description of the task.
4. Choose  *File*  or press `Ctrl` + `S` to save the entered description of the task.

Related Information

[Documentation \(Task/Process\) \[page 1112\]](#)

3.1.9.12.15 Using the Filter Editor

When specifying a filter to retrieve information from the identity store, you can use the filter editor to help you create a filter.

Context

When you define a filter, the SQL query will be automatically generated. To use the filter editor, proceed with the following steps:

Procedure

1. Choose the *Edit...* button, just above the right upper corner of the *Filter* field, to open the *Filter Editor* dialog box.
2. Specify the criteria by filling in the fields:

Option	Description
<i>Attribute Name</i>	The list contains all attributes in the identity store. Select one of the attributes. Note When you start typing, a selection dialog box will appear where you can browse for and select the attribute.
<i>Not</i>	Select <i>Not</i> to negate the expression in the <i>Filter</i> field.
<i>Filter</i>	Enter the selection criteria for the given attribute. This can be any valid expression according to the given database. You can add several values with a pipe () as the delimiter, e.g. Trondheim Oslo Bergen meaning Trondheim OR Oslo OR Bergen.

If you specify more than one attribute, the operator that is used by the filter editor to connect the attributes is *AND*.

3. Choose *OK* to close the dialog box.

The generated filter is inserted into the *Filter* field. It is possible to modify the generated filter, but it may not be possible to open the filter in the filter editor again.

Related Information

[Specifying the Conditions for a Switch Task \[page 251\]](#)

[Specifying the Conditions for a Conditional Task \[page 253\]](#)

[Specifying the Source for To-Passes \[page 327\]](#)

3.1.10 Maintaining Forms

Forms (previously known as UI tasks) are used to maintain identity store entries (users, roles, privileges, dynamic groups) in the Identity Management User Interface. A set of default forms is delivered as a separate package (com.sap.idm.forms.default) with the Provisioning Framework for SAP Identity Management 8.0. Forms are configured in the Identity Management Developer Studio.

A form contains the following:

- Attribute definitions: Which attributes are shown and in which order
- Access control: Who can execute the form
- Other user interface configuration elements, such as display name, header, footer

Depending on the form type, there are regular, display, search and access control forms. By default, forms are public objects. They are displayed on the [Public Objects](#) tab of a package. Forms can be removed from public and made public again.

In addition to the normal forms that can be accessed from the Identity Management User Interface, there are also a number of guided activities. These are intended to guide the user through more complex procedures. There are several types of guided activity forms available for different purposes:

- [Assignment Request](#)
An assignment request form is a guided activity that is used to request one or more roles or privilege assignments for a given user. This form type is used for context-based role and privilege assignments.
- [View Assignment Request](#)
A view assignment request form is a guided activity that is used to view the status of assignment requests. The form can also allow the user to restart failed or declined assignments.
- [Password Reset](#)
A password reset form is a guided activity that is used to enable users to reset their passwords from the Identity Management User Interface.

Each of the guided forms has various parameters that specify the options and behavior available to the user while performing the guided activity in the Identity Management User Interface.

Related Information

[Creating a Form \[page 271\]](#)

[Viewing and Configuring Form Properties \[page 274\]](#)

[Viewing and Configuring Guided Activity Form Properties \[page 275\]](#)

[Configuring the General Properties of a Form \[page 276\]](#)
[Configuring Result Handling for a Form \[page 277\]](#)
[Defining Attributes for a Form \[page 278\]](#)
[Copying and Pasting Rows in the Attributes Table of a Form \[page 279\]](#)
[Configuring Parameters for an Assignment Request Form \[page 283\]](#)
[Configuring Parameters for a View Assignment Request Form \[page 284\]](#)
[Configuring Parameters for a Password Reset Form \[page 285\]](#)
[Defining Form Access Control \[page 285\]](#)
[Configuring the Presentation of a Form \[page 286\]](#)
[Documenting a Form \[page 287\]](#)
[Deleting a Form \[page 288\]](#)

3.1.10.1 Creating a Form

Forms define the user interface which end users can use to modify or add data to the identity store.

Prerequisites

The package must be checked out and ready for editing.

Context

To create a form, proceed as follows:

Procedure

1. In the Identity Management Developer Studio tree view, select the *Forms* folder in the package where you want to create the form.
2. Choose *New* from the context menu.
3. You have the following options:
 - To create a Form folder, choose *Folder*.
 - To create a form, choose *Form*.
 - To create a guided activity form, choose *Assignment Request*, *View Assignment Request* or *Password Reset* form.

The form is created and added in the tree view. By default, forms (except for guided activity forms) are public objects. You can remove them from public or make them public again.

4. The properties of the form are automatically opened and displayed once the form is created. You can now start configuring the form.
5. You can modify the name of the form in the tree view. Select the form in the tree view and choose [Rename](#) from the context menu.

Related Information

[Viewing and Configuring Form Properties \[page 274\]](#)

[Viewing and Configuring Guided Activity Form Properties \[page 275\]](#)

[Form Properties \[page 1091\]](#)

3.1.10.2 Configuring Form Layout

You can configure the layout of a form that is displayed in the Identity Management User Interface.

Context

You can define form elements, such as tabs, columns, sections, lines and labels and arrange them in a way that you prefer. Configuring a form layout is valid only for normal forms and not for guided activities forms, such as [Assignment Request](#), [View Assignment Request](#) and [Password Reset](#).

To configure the layout of a form, proceed as follows:

Procedure

1. In the Identity Management Developer Studio tree view, select the form and choose [Properties](#) from the context menu.
2. Select the [Attributes](#) tab.
3. Select an entry type.
4. In the [Form Attributes](#) table, select [Insert Row](#) from the context menu and choose a form element. You have the following options:
 - [Label](#) - defines a label
 - [Line](#) - draws a line
 - [Section](#) - defines a section
 - [Column](#) - defines a column for attributes
 - [Tab](#) - defines a tab

ⓘ Note

The following rules apply when configuring a form layout:

When configuring a *Tab* element, you can only insert a *Section*, a *Label* or an attribute directly below the *Tab*. *Line* and *Column* elements cannot be inserted directly below the *Tab*. You can add them below any other element, for example a *Label*, or an attribute. See the screenshot.

The last element of a form is always an attribute.

All layout elements that violate these rules are silently ignored when the form is displayed in the Identity Management User Interface.

TestForm

General | Result Handling | **Attributes** | Access Control | Presentation | Documentation

Entry Type: 7/MX_PERSON ... ☐ Report Form
☐ This Form Creates a New Entry

Form Attributes:

Attribute	List	Mandatory	Read Only	Default Value/Caption
--Tab--				#MX_TASK_TESTFORM_CAPTION5
--Section--				#MX_TASK_TESTFORM_CAPTION8
--Label--				#MX_TASK_TESTFORM_CAPTION9
--Column--				
DISPLAYNAME	☒	☒	☐	
MSKEYVALUE	☒	☒	☐	
MX_ENTRYTYPE	☒	☒	☐	
MX_FIRSTNAME	☒	☐	☐	
MX_LASTNAME	☒	☐	☐	
--Line--				
MX_ADDRESS_CITY	☒	☐	☐	
MX_ADDRESS_COUNTRY	☒	☐	☐	
--Line--				
MX_PHONE_PRIMARY	☒	☐	☐	
MX_MOBILE_PRIMARY	☒	☐	☐	

Up
Down

① Note

If for some reason you cannot select *Insert Row* from the context menu, press Esc and try again.

- To provide a caption for *Label*, *Section*, and *Tab* form elements, in the *Default Value/Caption* column, choose the button.
- In the *Localize Text* dialog box, enter a caption and choose *OK*.

For example, name a tab **Test Tab** and name a section **Test Section**. Define one label: **Test Label**, as well as two lines and a column.

The form elements are added to the *Form Attributes* table. You can move them up or down with the *Up* and *Down* buttons to the right of the table or delete them by selecting *Delete Row* from the context menu.

You can also define attributes for this form in the Identity Management User Interface, described in *Defining Attributes for a Form*.

7. Choose *Save*.

The newly defined form elements are added to the form and displayed in the Identity Management User Interface.

TestForm Help

Unique ID: TestUser
Display Name: TestUser

Save Modify Refresh

Test Tab

Test Section

Test Label

Display Name: *

Unique ID: *

Entry Type: *

First Name:

Last Name:

City:

Country Key:

Primary Telephone Number:

Primary Mobile Number:

Related Information

[Defining Attributes for a Form \[page 278\]](#)

3.1.10.3 Viewing and Configuring Form Properties

Context

To view or configure the form properties, proceed as follows:

Procedure

1. Select the form that you want to view or configure the properties for in the tree view in Identity Management Developer Studio and choose *Properties* from the context menu

Note

If you just have created a form, its properties are opened automatically after creation.

2. Select a tab to enter or view the information about the form. The form properties contain the following tabs:

- [General](#)
- [Result Handling](#)
- [Attributes](#)
- [Access Control](#)
- [Presentation](#)
- [Documentation](#)

For more information and details about the tabs, properties and fields, see [Form Properties](#).

Note

The package must be checked out and ready for editing before changing the properties.

3. Choose  [File](#)  or press `Ctrl` + `S` to save the changes in the form properties.

Related Information

[Form Properties \[page 1091\]](#)

[Creating a Form \[page 271\]](#)

3.1.10.4 Viewing and Configuring Guided Activity Form Properties

Guided activity forms intended to guide the user through more complex procedures.

Context

Several types of guided activity forms are provided for different purposes:

- Assignment request
- View assignment request
- Password reset

To view or configure the guided activity form properties, proceed as follows:

Procedure

1. Select the form that you want to view or configure properties for in the tree view in Identity Management Developer Studio and choose [Properties](#) from the context menu.

Note

If you have just created a form, its properties are opened automatically during creation.

2. Select a tab to enter or view the information about the form. The following tabs are available in the guided activity form properties:

- [General](#)
- [Result Handling](#)
- [Parameters](#)
- [Access Control](#)
- [Presentation](#)
- [Documentation](#)

For more information and details about the tabs, and the properties and fields available, see Form Properties.

Note

The package must be checked out and ready for editing before you can change the properties.

3. Choose  [File](#)  or press `Ctrl` + `S` to save the changes in the form properties.

Related Information

[Form Properties \[page 1091\]](#)

[Creating a Form \[page 271\]](#)

3.1.10.5 Configuring the General Properties of a Form

Use the [General](#) tab of the form properties to enable/disable the form, define the form type or a repository for the form, and to add custom parameters.

Context

To configure the general properties of the form, proceed as follows:

Procedure

1. Select the form in the tree view.

2. Choose [Properties](#) from the context menu to open the form properties.
3. Select the [General](#) tab.
4. View the general information about the form or make changes to the properties. Enable or disable the form, define the form type or a repository, or add custom parameters. For more information about the fields and their options, see [General \(Form\)](#).
5. Choose  [File](#)  or press `Ctrl` + `S` to save the changes in the form properties.

Related Information

[General \(Form\) \[page 1091\]](#)

3.1.10.6 Configuring Result Handling for a Form

Use the [Result Handling](#) tab in the form properties to configure result handling for a form.

Context

To configure result handling, proceed as follows:

Procedure

1. Select the form in the tree view and choose [Properties](#) from the context menu.
2. Select the [Result Handling](#) tab.
3. View the result handling configuration or make changes.

Note

To be able to make changes in the configuration, the package has to be checked out and ready for editing.

For details about the fields on the [Result Handling](#) tab, see [Result Handling \(Form\)](#).

4. Choose  [File](#)  or press `Ctrl` + `S` to save any changes.

Related Information

[Result Handling \(Form\) \[page 1093\]](#)

3.1.10.7 Defining Attributes for a Form

Use the [Attributes](#) tab to configure which attributes are displayed for a form in the Identity Management User Interface.

Context

To define the attributes for a form, proceed as follows:

Procedure

1. Select the form in the tree view and choose [Properties](#) from the context menu.
2. Select the [Attributes](#) tab.
3. Enter the following values in the fields:
 - To select the entry type for this form, choose the ... button.
To specify that this is a form used to generate reports, select the [Report Form](#) check box.
 - To specify that this form is used to create a new entry in the identity store, select the [This Form Creates a New Entry](#) check box.

For more information about the fields and their options, see [Attributes \(Form\)](#)

4. The list contains all attributes defined for the selected entry type. You can perform the following operations:
 - Select [List](#) to specify that the attribute is to be included in the Identity Management User Interface.
 - Select [Mandatory](#) to specify that the attribute must be filled in before the form can be executed.
 - Select [Read Only](#) to specify that this attribute is read-only in the web interface.
 - Enter a default value for the attribute. This default value will be used as follows:
Default values are only used if the attribute does not already contain a value.
If the attribute is visible, the default value will be filled in when the form is displayed. If the attribute is invisible, the default value will be stored in the identity store without being displayed. This can be used to set system attributes or other attributes that should not be visible to the user.
You can also combine default values with read-only attributes. The default value is then displayed in the web interface and stored in the identity store, but the user cannot modify the attribute.
5. To add an attribute to the list, go to a new row and press Ctrl + Space. This opens a dialog box with all attributes allowed for this entry type.
6. Specify the order to list attributes in the web interface by moving the attribute up and down in the list with the [Up](#) and [Down](#) buttons to the right of the list. You can also move the attributes up or down by dragging the handle to the left of the attribute name with the mouse. The attributes are displayed in the same order as they are here.
7. Choose File Save or press Ctrl + S to save your changes.

Related Information

[Attributes \(Form\) \[page 1094\]](#)

3.1.10.8 Copying and Pasting Rows in the Attributes Table of a Form

You can copy and paste multiple rows on the [Attributes table](#) of a form.

Prerequisites

The package is checked out.

Note

When the package is checked in, you can only copy rows.

Context

When managing the rows of the [Attributes table](#), the following options are available:

- Context menu – Allows you to copy and paste one or multiple rows, insert a row, and delete rows.
- Multiselect – Allows you to select multiple rows using the mouse and **Ctrl** or **Shift** keys.
- Drag and drop – Allows you to move one or more rows up and down within the [Attributes table](#).

To manage the rows in the [Attributes table](#) of a form, proceed as follows:

Procedure

1. Select the form that you want to manage.
2. Select the [Attributes table](#) and choose the option you want to use from the context menu.

Depending on the steps you are executing, there are two different context menus available:

- SAP Identity Management context menu - It is opened by right-clicking the Attributes table without selecting a cell.
- Built-in context menu with Eclipse options for the cell - It is opened when you left-click to select a cell from the [Attributes table](#) and right-click it.

📘 Note

You cannot copy and paste columns.

📘 Note

When you copy information from the *Attributes table* to an external table source, an additional column will appear. It contains the codes of all used visualization elements.

Option	Description
Copy Selected Rows	Select one or more rows in the <i>Attributes table</i> and choose <i>Copy Selected Rows</i> from the context menu. You can also copy empty rows. When the copied rows are pasted in an external spreadsheet, the rows are separated by new lines and the columns are separated by tabs.
Paste Rows	Position your mouse cursor at the row of your choice and choose <i>Paste Rows</i> from the context menu. You can also paste empty rows. When the copied rows are pasted in the <i>Attributes table</i> of a form, expect the following: • Pasted rows overwrite the existing ones and does not add new rows. • Pasted rows at the end of the table expand it automatically. 📘 Note The selection of the <i>Paste Rows</i> option triggers a validation whether the <i>Attributes table</i> of the form contains the attributes that you are pasting. A notification dialog informs you for each attribute that already exists in the table and will not be pasted.

Option	Description
Insert Row	Position your mouse cursor at the row of your choice and choose <i>Insert Row</i> from the context menu. The submenu contains the following items: • <i>Label</i> • <i>Line</i> • <i>Section</i> • <i>Column</i> • <i>Tab</i> • <i>Attribute</i> The selection of the <i>Attribute</i> option adds an empty row, where you can type the name of an existing attribute and add it to the form.  Note If the <i>Attributes table</i> of the form already contains the attribute that you are trying to add, a notification dialog informs you that it will not be inserted.
Delete Selected Rows	Select one or more rows in the <i>Attributes table</i> and choose <i>Delete Selected Rows</i> from the context menu.

3.1.10.9 Configuring Form Specific Attributes

You can configure the attributes that are defined for a form. As a result, all configured attributes become form specific attributes.

Context

Once you have defined the attributes for a form, you can configure some of the attribute properties. These attribute properties have default values that are delivered with the identity store schema. When you change the default values of the attribute properties, you turn the standard schema attributes into form specific attributes. This means that your changes are valid only for this particular form. Any updates in the identity store schema attributes will not overwrite the form specific attributes of this form.

To configure the form specific attributes, follow the steps below:

Procedure

1. In the tree view, select the form which attributes you want to configure and choose *Properties* from the context menu.
2. Select the *Attributes* tab.
3. Select an attribute and choose the button next to it to open the attribute properties.
4. Select a tab and configure the attribute properties. You can configure only the following properties:

Tab	Property	Description
<i>Presentation</i>	<i>Confirm Input</i>	See Presentation (Attribute) [page 992] .
	<i>Hide Input</i>	See Presentation (Attribute) [page 992] .
<i>Assignment</i>	All properties can be configured.	See Assignments (Attribute) [page 996] .
<i>Validation</i>	<i>Regex</i>	See Validation (Attribute) [page 998] .
<i>Attribute Values</i>	All properties can be configured.	See Attribute Values (Attribute) [page 999] .

Note

To avoid inconsistent behavior of legal attribute values, make sure that the attribute value options defined on the default level (Identity Store Schema) and on the form specific attribute level do not differ.

For example, if the default attribute values are defined with an SQL query, they should also be defined with an SQL query on the form specific attribute level.

5. Choose *OK*.

Results

When you change the attribute properties, *Form specific* is added to the name of the corresponding tab. For example:

The screenshot shows a configuration window titled "Identity Store Attribute -DISPLAYNAME". It has several tabs: "General", "Storage", "Presentation (Form s)", "Validation (Form spe", "Event Processes", "Attribute Values (Fo", and "Entry Types". The "Presentation (Form s)" tab is selected. Inside this tab, there is a sub-header "Presentation (Form specific)". Below this, the following fields and options are visible:

- Display Name:** A text field containing "#MX_DISPLAYNAME_DN" and a button with three dots.
- Tooltip Text:** A text field and a button with three dots.
- Presentation:** A dropdown menu currently showing "SingleLine".
- Options:**
 - ☐ Confirm Input
 - ☐ Hide Input
 - ☐ Show Search Field
 - ☒ List Entries on Load
- Number of Lines:** A text field containing the value "5".
- Task Defaults:**
 - ☐ Read Only
 - ☐ Mandatory

At the bottom right of the dialog are "OK" and "Cancel" buttons.

3.1.10.10 Configuring Parameters for an Assignment Request Form

Use the [Parameters](#) tab to configure the specific parameters for an assignment request form.

Context

To configure the assignment request form parameters, proceed as follows:

Procedure

1. Select the form in the tree view and choose [Properties](#) from the context menu.

2. Select the *Parameters* tab.
3. View or change the assignment request parameters. You can make various settings, such as selecting the entry type for this form, specifying if an assignment has to include a context, and selecting which entry type (role or privilege) will be added by this form. For more information about the fields and their options, see Parameters (Assignment Request Form).
4. Choose ► *File* ► *Save* or press **Ctrl** + **S** to save the changes in the form properties.

Related Information

[Parameters \(Assignment Request Form\) \[page 1095\]](#)

3.1.10.11 Configuring Parameters for a View Assignment Request Form

Use the *Parameters* tab to configure the specific parameters for a view assignment request form.

Context

To configure the view assignment request form parameters, proceed as follows:

Procedure

1. Select the form in the tree view and choose *Properties* from the context menu.
2. Select the *Parameters* tab.
3. View or change the view assignment request parameters. You can perform various operations, including selecting the entry type for this form and specifying that the form should include the option to restart failed or declined privilege assignments. For more information about the fields and their options, see Parameters (View Assignment Request Form).
4. Choose ► *File* ► *Save* or press **Ctrl** + **S** to save the changes in the form properties.

Related Information

[Parameters \(View Assignment Request Form\) \[page 1097\]](#)

3.1.10.12 Configuring Parameters for a Password Reset Form

Use the [Parameters](#) tab to configure the specific parameters for a password reset form.

Context

To configure the password reset form parameters, proceed as follows:

Procedure

1. Select the form in the tree view and choose [Properties](#) from the context menu.
2. Select the [Parameters](#) tab.
3. View or change the password reset parameters. You can make various settings here, including selecting the entry type for this form, and specifying how many password questions the user has to answer. For more information about the fields and their options, see [Parameters \(Password Reset Form\)](#).
4. Choose  [File](#)  or press `Ctrl` + `S` to save the changes in the form properties.

Related Information

[Parameters \(Password Reset Form\) \[page 1097\]](#)

3.1.10.13 Defining Form Access Control

Use the [Access control](#) tab to specify who should be allowed to access this form from the Identity Management User Interface. To do this, specify one or more access control rules defining the users for whom this form should be available. You can also provide anonymous access to a form. This is the case for a password reset form or a self-registration form.

Context

To define access control for a form, proceed as follows:

Procedure

1. Select the form in the tree view.
2. Choose [Properties](#) from the context menu to open the form properties.
3. Select the [Access Control](#) tab.
4. You can perform the following operations:
 - To define an access control rule, choose [Add](#).
 - To modify an access control rule, select the rule from the list and choose [Modify](#).
 - To remove an access control rule, select the rule from the list and choose [Remove](#).

For details about the fields on the [Access Control](#) tab, see [Access Control \(Form\)](#).

5. Choose  [File](#) > [Save](#)  or press Ctrl + S to save any changes that you have made.

Related Information

[Access Control \(Form\) \[page 1100\]](#)

3.1.10.14 Configuring the Presentation of a Form

Use the [Presentation](#) tab to configure how the form is displayed in the Identity Management User Interface.

Context

To configure how the form is displayed, proceed as follows:

Procedure

1. Select the form in the tree view.
2. Choose [Properties](#) from the context menu to open the form properties.
3. Select the [Presentation](#) tab.

4. View the presentation configuration or make changes. Enter a display name for the form in the Identity Management User Interface, specify that the form should have a submit button, modify button, delete button.

For details about the fields on the [Presentation](#) tab, see Presentation (Form).

5. Choose **File > Save** or press **Ctrl** + **S** to save any changes that you have made.

Related Information

[Presentation \(Form\) \[page 1105\]](#)

3.1.10.15 Documenting a Form

Use the [Documentation](#) tab to enter a description of the form. This description is used in the system report.

Context

To enter a description of the form, proceed as follows:

Procedure

1. Select the form in the tree view.
2. Choose [Properties](#) from the context menu to open the form properties.
3. Select the [Documentation](#) tab.
4. Enter a text describing the form.
5. Choose **File > Save** or press **Ctrl** + **S** to save the changes in the form properties.

Related Information

[Documentation \(Form\) \[page 1108\]](#)

3.1.10.16 Deleting a Form

You can delete forms that you do not need any more.

Context

To delete a form, proceed as follows::

Procedure

1. In the tree view in Identity Management Developer Studio, select and expand the *Forms* folder from a package in your identity store.

Note

Make sure that the package is checked out and ready for editing.

2. Select a form that you want to delete and choose *Delete* from the context menu.

3.1.10.17 Copying a Form

You can copy a form from one package to another within the same identity store. To copy a form from one identity store to another, you need to export the package containing that form and import it to the identity store you want.

Prerequisites

- The package that you want to copy the form to is checked out.

Context

To copy a form within the same identity store, proceed as follows:

Procedure

1. In the tree view, select the form and choose [Copy](#) from the context menu.
2. Select the package that you want to copy the form to.
3. Select the [Forms](#) folder and choose [Paste](#) from the context menu.
4. Choose [Save](#).

Note

If the form that you want to copy to another package is referenced from a process, make sure the process is made public. If not, copy to will not be allowed. You will get a [Move Blocked](#) warning that the referenced object is not public.

3.1.11 Maintaining Jobs

3.1.11.1 Creating a Job

Jobs are stored and run within an identity store. They are added to a job folder in a package.

Prerequisites

- You have checked out the package.

Context

To create a job, proceed as follows:

Procedure

1. In the SAP Identity Management Developer Studio tree view, select the [Jobs](#) folder of the package.
2. Choose [New](#) [Job](#) from the context menu.

This will create a new job and open it in edit mode in the job perspective.

3. Change the name of the job in the tree view and make sure that the job is enabled, and a dispatcher selected. If required, you can change other job properties.
4. To save the changes in the job properties, choose  [File](#)  [Save](#) .

Next Steps

Continue by adding passes to the job.

Once you are ready, you can run the job either in the SAP Identity Management Developer Studio or in the SAP Identity Management Administration User Interface.

It is also possible to import jobs into the configuration. The imported jobs will then be inserted into the configuration as a new package in the [Packages](#) folder in the tree view.

Related Information

[Job Properties \[page 1022\]](#)

[Jobs and Passes \[page 78\]](#)

[Adding a Pass to a Job \[page 308\]](#)

[Running a Job \[page 297\]](#)

3.1.11.2 Creating a Repository Job

A repository job is a job that runs with a given repository. Each repository job has a defined repository type.

Prerequisites

- You have checked out the package.
- There is a repository type that you want to create a job for.

Context

To create a repository job, proceed as follows:

Procedure

1. In the SAP Identity Management Developer Studio tree view, select the *Jobs* folder of the package.
2. Choose ► *New* ► *Repository Job* ▾ from the context menu.
3. Select the repository type to be used for this job and choose *OK*.

Note

You cannot change the repository type once it has been set.

The repository job is created and open in edit mode in the job perspective. A list of all repositories of the selected repository type is displayed.

4. Change the name of the job in the tree view and make sure that the job is enabled, and a dispatcher selected. If required, you can change other job properties.
5. To save the changes in the job properties, choose ► *File* ► *Save* ▾.

Next Steps

Continue by adding passes to the job.

Once you are ready, you can run the job either in the SAP Identity Management Developer Studio or in the SAP Identity Management Administration User Interface.

Related Information

[Adding a Pass to a Job \[page 308\]](#)

[Running a Job \[page 297\]](#)

3.1.11.3 Viewing and Configuring Job Properties

Context

To view or configure the job properties, do the following:

Procedure

1. Select the job in the tree view and choose *Properties* from the context menu.
2. Select a tab to enter or view the information about the job:
 - *General*
 - *Logging*
 - *State*
 - *Documentation*
3. Choose  *File*  to save any changes made to the job properties.

Related Information

[Job Properties \[page 1022\]](#)

[General \(Job\) \[page 1023\]](#)

[Logging \(Job/Action Task\) \[page 1024\]](#)

[State \(Job/Action Task\) \[page 1025\]](#)

[Documentation \(Job\) \[page 1027\]](#)

[Searching for an Object in the Configuration \[page 394\]](#)

3.1.11.4 Entering a Description of a Job

You can enter a description that is used as the internal documentation for a job.

Context

Note

This documentation is only used for development purposes, and will never be visible to other users.

To document the job, proceed as follows:

Procedure

1. Select the job that you want to document in the tree view in Identity Management Developer Studio.
2. Choose *Properties* in the context menu to view the job properties.
3. Select the *Documentation* tab.
4. Enter the description of the job.
5. Choose ► *File* ► *Save* or press **Ctrl** + **S** to save your job description.

Related Information

[Documentation \(Job\) \[page 1027\]](#)

[Job Properties \[page 1022\]](#)

3.1.11.5 Enabling or Disabling Jobs

To specify if a job should be run or not, it is enabled or disabled.

For a job to be executed, it must be enabled for processing, and have a dispatcher defined that is responsible for running it.

You can enable or disable jobs in the following ways:

- Enable or disable a single job
- Enable or disable all jobs in a folder

Related Information

[General \(Job\) \[page 1023\]](#)

[Enabling or Disabling All Jobs in a Folder \[page 294\]](#)

[Enabling or Disabling a Single Job \[page 294\]](#)

3.1.11.5.1 Enabling or Disabling a Single Job

You can enable or disable a single job.

Context

To enable or disable a job

Procedure

1. Select the job in the tree view and choose *Properties* from the context menu.
2. Select the *General* tab.
3. Select or deselect the *Enabled* check box.
4. Choose ► *File* ► *Save* ◄ or press Ctrl + S to save the changes made to the job properties.

Related Information

[Job Properties \[page 1022\]](#)

[General \(Job\) \[page 1023\]](#)

3.1.11.5.2 Enabling or Disabling All Jobs in a Folder

You can enable or disable all jobs in a folder.

Context

To enable or disable all jobs in a job folder:

Procedure

1. Select the job folder in the tree view.
2. Choose *Enable sub-tree* or *Disable sub-tree* from the context menu.

3.1.11.6 Scheduling a Job

Schedule rules define when and how often a job is run.

Prerequisites

To run a job, a dispatcher needs to be defined for the job.

To schedule a job you must either check out the package or leave it checked-in, but in this case, you need to have assigned the Developer or the Owner role for the package.

Context

When a job is created, a default schedule rule is defined. You can change this if necessary.

Procedure

1. Select the job in the tree view and choose [Properties](#) from the context menu.
2. Select the [General](#) tab.
3. Select one of the predefined schedule rules in the [Schedule rule](#) list.
4. Choose [Save](#).

Results

The [Scheduled time](#) field displays the next time the job is scheduled to run.

Related Information

[General \(Job\) \[page 1023\]](#)

[State \(Job/Action Task\) \[page 1025\]](#)

[Providing Authorizations to a Package \[page 196\]](#)

[Authorized Users \(Packages\) \[page 1009\]](#)

3.1.11.7 Selecting a Dispatcher for a Job

Before a job can be run, a dispatcher needs to be defined.

Prerequisites

To select a dispatcher for a job you must either check out the package or leave it checked-in, but in this case, you need to have assigned the Developer or the Owner role for the package.

Context

Select the dispatcher that is allowed to run the job.

Procedure

1. Select the job in the tree view and choose *Properties* from the context menu.
2. Select the *General* tab.
3. Select one or more dispatchers from the *Run by dispatchers* list.
4. Choose *Save*.

Related Information

[General \(Job\) \[page 1023\]](#)

[Providing Authorizations to a Package \[page 196\]](#)

[Authorized Users \(Packages\) \[page 1009\]](#)

3.1.11.8 Configuring a Job's Logging

You can change the log level and other log options for a job.

Context

A job is created with a set of default values for the log options. If necessary, you can change these later:

Procedure

1. Select the job in the tree view and choose *Properties* from the context menu.
2. Select the *Logging* tab.
3. Change the log level or other options for the job log.
4. Optionally, select a script that will be called if the job fails.
5. Choose "Save".

Related Information

[Logging \(Job/Action Task\) \[page 1024\]](#)

[Viewing the Job Log \[page 300\]](#)

3.1.11.9 Running a Job

When developing a job, you can run it either in SAP Identity Management Developer Studio or in SAP Identity Management Administration User Interface.

Note

In a production environment, jobs are normally not started or stopped manually, except for initial load and reconciliation jobs for repositories that are started in the SAP Identity Management Administration User Interface.

There are several ways of running jobs (both repository jobs and regular jobs) when developing solutions. When you start the job, the dispatcher picks up the job and starts the runtime engine that processes it.

Running Repository Jobs

- To run a repository job in SAP Identity Management Developer Studio, choose one of the following options:
 - Select the job in the tree view and choose [Run with Repository](#) from the context menu. Select a repository and choose [OK](#).
 - Open the job in the job view and choose the  (Run) button. Select a repository and choose [OK](#).
 - Open the job properties and select the [General](#) tab. Select a repository and choose [Run](#) from the context menu.
- To run a repository job in SAP Identity Management Administration User Interface, configure the following:
 1. In the SAP Identity Management Developer Studio tree view, open the repository type that you created the job for and select the [General](#) tab.
 2. Add a new repository type constant to the list by choosing [New](#) from the context menu.
 3. In the [Repository Type Properties](#) dialog box, select [Job Reference](#) for the [Data Type](#) field and then select the repository job to be the value of this repository type constant. Choose [OK](#).
 4. Save your changes.
 5. To run the repository job from the SAP Identity Management Administration User Interface, choose [System Configuration](#) > [Repositories](#) > [Repository](#) > [Jobs](#) , select the job and choose [Run Now](#).

Running Regular Jobs

- To run a regular job in the SAP Identity Management Developer Studio, choose one of the following options:
 - Select the job in the tree view and choose [Run](#) from the context menu.
 - Open the job in the job view and choose the  (Run) button.
 - Open the job properties and select the [General](#) tab. Choose [Run](#).
 - Open the job properties and select the [State](#) tab. Choose [Run](#).
- To run a regular job in SAP Identity Management Administration User Interface, configure the following:
 1. In the SAP Identity Management Developer Studio tree view, select the package where you have created your job.
 2. Select [Constants](#) and choose [New Package Constant](#) from the context menu.
 3. In the [Package Constant](#) dialog, select [Job Reference](#) for the [Data Type](#) field and then select the job to be the value of this package constant. Choose [OK](#).

Note

Make sure the [Show in the Admin User Interface](#) checkbox is selected.

4. Save your changes.
5. To run the job in the SAP Identity Management Administration User Interface, choose [System Configuration](#) > [Packages](#) > [Package](#) > [Jobs](#) , select the job and choose [Run Now](#).

Related Information

[Jobs and Passes \[page 78\]](#)

3.1.11.9.1 Testing a Job

When developing a job, you can test it by running it in the SAP Identity Management Developer Studio.

Prerequisites

- You have created and configured a job, including passes.
- The dispatcher is running.

Context

To test a job, proceed as follows:

Procedure

1. Run the job using one of the following methods:
 - Select the job in the tree view and choose *Run* from the context menu.
 - Open the job and choose *Run* from the context menu.
 - View the properties of the job and choose *Run* on the *General* tab.
 - View the properties of the job and choose *Run* on the *State* tab.
2. The job log is automatically opened when you run the job. You can see if it completes successfully or check any errors in the log.
3. You can control job execution in one of the following ways:
 - If the job runs for a long time, you can signal the job to be stopped by choosing *Stop now* on the *General* or *State* tab.
 - If the job fails, you can force a restart of the job by choosing *Force restart* on the *State* tab.

Related Information

[General \(Job\) \[page 1023\]](#)

[State \(Job/Action Task\) \[page 1025\]](#)

[Viewing the Job Log \[page 300\]](#)

3.1.11.9.2 Viewing the Job Log

The job log displays information about the execution of the job. Each line in the log shows information about one execution of the job. How many log lines are kept depends on the configuration of the scheduled procedure that is responsible for clearing old log entries.

Prerequisites

- You have run the job at least once.

Context

The job log can be viewed at the following levels:

- The global job log shows all jobs in the system.
- The package job log shows all jobs in the package.
- The job log at the job level shows the log entries for that specific job.

To view the job log:

Procedure

1. Open the job log in one of the following ways:
 - When you run a job, the job log is displayed automatically.
 - Select the job in the tree view and choose *Job log* from the context menu.
 - If the job is open in the job view, choose *Job log* from the menu in the job view.
 - To view the package job log, select the package's *Job log* entry in the tree view and choose *Open* from the context menu.
 - To view the global job log, select the package's *Job log* entry under the *Management* node in the tree view and choose *Open* from the context menu.

The job log opens in the log view.

Column	Description
Status indicator	The first column in the log contains a symbol indicating the status of the job. There are symbols corresponding to each status.
<i>Job</i>	The ID and name of the job.  Note This column is not visible on the job level.
<i>Date/Time</i>	The date/time when the job was run. If the job is in progress, this will be the last time the log was refreshed.
<i>Status</i>	The status can be one of the following values: <i>OK</i> The job has completed without errors. <i>Warning</i> The job has completed with warnings. <i>Warning and Error</i> The job has completed with both warnings and errors. <i>Error</i> The job has completed with errors. <i>Running</i> The job is in process. <i>Stopping</i> The job has been signaled to stop.
<i>Duration</i>	The duration of the run in seconds. For a running job, this is how long the job has been running.
<i>Total entries</i>	The total number of entries processed in this run. For a running job, this is the number of entries processed so far.
<i>Added</i>	The number of entries added in this run. Empty for a running job.
<i>Modified</i>	The number of entries modified in this run. Empty for a running job.
<i>Deleted</i>	The number of entries deleted in this run. Empty for a running job.

Column	Description
<i>No operation</i>	The number of entries that were not affected. Empty for a running job.
<i>Repository</i>	The repository towards which the job has been run, if any.
<i>Package</i>	The package to which the job belongs.
 Note Only visible in the global job log.	

2. You can do the following:

- a. Refresh the log by choosing *Refresh* from the log view menu or by entering a refresh interval.
- b. Filter job logs by specifying the date range in the *Start Date* and *End Date* fields from the log view menu. Filtering logs by date speeds up the process of searching for specific logs.

 Note

The default value of the *Start Date* and *End Date* fields from the job log menu is the current date. Any changes in the specified date range are retained as long as the job view stays opened. If you close it, the changes will be retained by the end of the current day.

- c. Select which log messages to include in the list by choosing *Show* and selecting *All*, *Warnings and Errors* or *Errors* from the log view menu.
- d. Export the log to a .CSV file by choosing *Export log* from the log view menu.
- e. Delete the job log by choosing *Delete log* from the log view menu.

Confirm that you want to delete the log.

- f. Customize the width of the columns in the job log view.

 Note

The columns width is retained when the job log is reopened.

3. View the details on one specific run by selecting the line and choosing *Log details* from the context menu.

The detailed log for the log entry is opened. The log contains information about this specific job execution, including the version of the runtime engine, the job GUID and information about each pass in the job. You can also view the start time and the end time of the job run.

Start time 26.05.2021 10:13:45

End time 26.05.2021 10:13:48

Product	Identity Center
Version	DSE.JAR version: 8.0.9.0 Built: 09.05.2021 18:08:34 - 243396 (c) Copyright 2012 SAP AG. All rights reserved.
Customer	SAP customer : f9c1c5cd66189d133765ac44ea6c127a
Job GUID	54CC39B5-4D80-437B-8AB4-BD97EA8FF049
Job Source	Identity Management JobID:1
Job Name	Create Test Data
Dispatcher	sybase
Work area	c:/usr/sap/ldm/Identity Center/Jobs/54CC39B5-4D80-437B-8AB4-BD97EA8FF049
Time used	3
Entries added	1
Entries modified	25
Entries deleted	0
Total entries	26
Errors	1
Warnings	0

- a. Choose [Save log](#) to save the log entry to a text file.
Add a file name and location for the file and choose [Save](#).

Related Information

[Running a Job \[page 297\]](#)

3.1.11.9.3 Viewing the Execution Log

The execution log shows log information from scripts that are called from jobs or tasks or from Java extensions that are part of frameworks.

Prerequisites

- You have run a job with a script that write log entries with uInfo, uWarning or uError.
- You are running tasks in the SAP Provisioning Framework Version 2.

Context

To view the execution log:

Procedure

1. To view the execution log, select the package's *Execution log* entry under the *Management* node in the tree view and choose *Open* from the context menu.

The job log opens in the log view. The log is limited to the number of entries specified for the Identity Center.

Each log entry contains the following information:

Column	Description
Level	A symbol indicating log level (Info, Warning, Error).
<i>Component</i>	The job or task's ID and name.
<i>Type</i>	Component type. (Job, Task, Java extension).
<i>Date</i>	The date/time when the log entry was written.
<i>Audit ID</i>	Audit ID related to this log entry.
<i>Entry</i>	The MSKEYVALUE of the entry that was processed.
<i>Repository</i>	The repository towards which the job has been run, if any.
<i>Message</i>	The text that was written from the component.

2. You can do the following:
 - a. Refresh the log by choosing *Refresh* from the log view menu or by entering a refresh interval.

Note

The selected automatic refresh option and the specified refresh interval in the log view menu are retained when the execution log is reopened.

- b. Customize the width of the columns in the execution log view.

Note

The columns width is retained when the execution log is reopened.

- c. Select which log messages to include in the list by entering a text in the *Filter* field above the search result. Only the lines in the current search result matching the filter are displayed. You can filter on the *Component*, *Date*, *Entry*, *Repository* and *Message* columns.
- d. You can enter a search criteria in the following way:

- Choose [Search options](#) from the context menu.
- Fill in the fields [Job ID](#), [Task ID](#), [Audit ID](#) or [MSKEYVALUE](#) to only display log entries that match your selection.
- Choose [OK](#) to search for log entries matching the search criteria.

The log entries corresponding to the search criteria will be displayed in the list.

- e. You can open the properties of the job or task by selecting the log entry and choosing [Open job](#) or [Open action task](#) from the context menu.
- f. You can export the log to a CSV file:
 - Choose [Export log](#) from the context menu.
 - Enter a file name and location of the CSV file.
 - Choose [Save](#).

3.1.11.10 Running Jobs in Sequence

You can run jobs in sequence. This is valid both for jobs and repository jobs.

Prerequisites

- You have a package that contains at least two jobs.
- The package is checked out and ready for editing.

Context

To run jobs in sequence, proceed as follows:

Procedure

1. Select the job that you want to run first and choose [Open](#) from the context menu.
2. Select the [Passes](#) tab and choose [New](#) [From Custom](#) pass.

This pass will be used to call the next job by jobID.

To modify the name of the pass, choose [Rename](#) from the context menu and enter **Scheduling Next Job** pass, for example.

3. In the jobs view, select the [Scripts](#) tab and choose [New](#) from the context menu.
4. In the [New Script](#) dialog box, enter a name for the script and choose [OK](#).

This script uses the [uRunJobNow](#) function.

5. In the script editor view, enter the following script code:

- For jobs, enter:

Sample Code

```
function scheduleNextJob(Par) {  
    var q = "select JobId from mc_jobs where name like 'Second Initial  
load'";  
    var jobid = uSelect(q);  
    uRunJobNow(jobid);  
}
```

- For repository jobs, enter:

Sample Code

```
function scheduleNextJob(Par) {  
    var RepId = uGetRepositoryID();  
    var RepName = uGetRepositoryName(RepId);  
    var q = "select JobId from mc_jobs where name like 'Initial load  
["+RepName+"]' and Repository = '"+RepName+"'";  
    var jobid = uSelect(q);  
    uRunJobNow(jobid);  
}
```

6. Choose [Save](#).
7. Open the [Scheduling Next Job](#) pass and select the [General](#) tab.
8. In the [Termination Script](#) field, select the script you have created and choose [Save](#).
9. Run the job.

Note

You need to configure the steps above for every job that you want to run in sequence. For example, to run ten jobs in sequence, repeat the steps above for the first nine jobs.

Results

Running jobs in sequence is now configured. Regardless of whether the first job completed successfully or not, the next job is called.

3.1.11.11 Viewing a Job's State

Viewing the state of a job or force a failed job to run again.

Context

When you configure a job, you use the [State](#) tab to view the state of the job.

Procedure

1. Select the job in the tree view and choose [Properties](#) from the context menu.
2. Select the [State](#) tab.
3. View the state properties of a job. You can also run or stop a job, or force a restart of a stalled job.
4. Choose [Refresh](#) to update the information.

Related Information

[Testing a Job \[page 299\]](#)

[State \(Job/Action Task\) \[page 1025\]](#)

3.1.11.12 Maintaining Passes

Passes are part of a job, where each pass in a job is responsible for performing a specific operation. Maintaining passes includes the operations like creating the passes, viewing and configuring the pass properties, and deleting the passes.

Related Information

[Pass Properties \[page 1029\]](#)

[Jobs and Passes \[page 78\]](#)

3.1.11.12.1 Adding a Pass to a Job

The passes of a job specify how data should be read from or written to a number of repositories.

Prerequisites

- The package is checked out.
- The job is opened in the job perspective.

Context

To add a pass to a job:

Procedure

1. Select the [Passes](#) tab in the job view.
2. Choose [New](#) from the context menu and select one of the available pass types.
3. Rename the pass in the pass list.
4. Configure the pass properties in the pass editor.
5. Choose  [File](#)  or press `Ctrl` + `S` to save the changes.

Related Information

[Jobs and Passes \[page 78\]](#)

[Pass Properties \[page 1029\]](#)

3.1.11.12.2 Viewing and Configuring Pass Properties

Prerequisites

- Make sure that the package is checked out before editing.

- The job is opened in the job perspective.

Context

To view or configure the pass properties, do the following:

Procedure

1. Select the *Passes* tab in the job view.
2. Select the pass you want to view or configure and choose *Open* from the context menu.
3. Select a tab to enter or view the information about the pass in the pass editor.
 - *General*
 - *Source*
 - *Destination*
 - *Delta*
 - *Documentation*
4. Choose ► *File* ► *Save* ◄ or press `Ctrl` + `S` to save the changes.

Related Information

[Pass Properties \[page 1029\]](#)

3.1.11.12.3 Copying and Moving Pass Rows

You can copy and paste multiple rows on the *Destination* tab of all passes and the *Source* tab of the From Custom pass. This way you can manage the content of the rows in passes or in external table sources (for example, Microsoft Excel) and paste it wherever needed.

Prerequisites

The package is checked out.

Note

When the package is checked in, you can only copy rows.

Context

When managing pass rows, the following options are available:

- Context menu – Allows you to copy and paste one or multiple rows, insert a row, delete rows, and undo row deletion.
- Multiselect – Allows you to select multiple rows using the mouse and **Ctrl**/**Shift** keys.
- Drag and drop – Allows you to move one or more rows up and down within the definitions table of the pass using the left mouse button.

To manage pass rows in the Definition table of a pass, proceed as follows:

Procedure

1. Select the [Passes](#) tab in the job view.
2. Select the pass you want to manage.
3. Select the [Destination](#) tab and choose the option you want to use from the context menu. There are two ways to open the context menu. One is to right-click the definitions table without selecting a cell; the other is to left-click a cell to select it and then right-click the definitions table.

Note

You cannot copy and paste columns.

Note

When you copy and paste rows from external table source, the rows that you paste must have the exact matching columns in the definitions table of the pass. If they are more than expected, the columns exceeding the limit will be truncated.

Option	Description
Copy Selected Rows	Select one or more rows in the pass table and choose Copy Selected Rows from the context menu. You can also copy empty rows. When copied rows are pasted in external spreadsheet, rows are separated by new lines and columns are separated by tabs.

Option	Description
Paste Rows	Position your mouse cursor at the row of your choice and choose Paste Rows from the context menu. You can also paste empty rows. When copied rows are pasted in the definitions table of a pass, expect the following: • Pasted rows overwrite existing ones and does not add new rows. • Pasted rows at the end of the table expand it automatically. • It is checked whether the first column of the pasted rows contains valid prefixes, such as: (@), (#), (.) and others. Invalid values are replaced with empty strings.
Insert Row	Position your mouse cursor at the row of your choice and choose Insert Row from the context menu. You can insert an empty row.
Delete Selected Rows	Select one or more rows in the pass table and choose Delete Selected Rows from the context menu.
Undo Row Deletion	You can undo multiple row deletion steps regardless of whether you have saved or not your changes. You cannot undo your row deletion steps only if you close the pass properties.

4. Choose ► [File](#) ► [Save](#) or press **Ctrl** + **S** to save the changes.

3.1.11.12.4 Actions with Job Passes

You can move and reorder one or multiple passes in the [Job Folder](#) using drag and drop or the arrow keys on the top-right of the [Pass Folder](#). You can also execute any of the actions copy, cut, paste, delete, or enable on one or multiple selected passes.

Prerequisites

The package is checked out.

Context

When managing passes in the [Pass Folder](#), the following options are available:

- [Context](#) menu - Allows you to rename one selected pass or cut, copy, paste, enable, delete or move up or down one or multiple passes.
- Multiselect - Allows you to select multiple passes using the left mouse button and `Ctrl` or `Shift` keys.
- Arrows Up and Down - Allows you to move one or more passes one position up or down from their current position within the [Passes table](#).
- Drag and drop - Allows you to move one or more passes one or multiple position up or down within the [Passes table](#) of the job to the wanted position using the left mouse button.

To manage passes in the [Pass Folder](#) of the job, proceed as follows:

Procedure

1. Select the [Passes](#) tab in the [Job](#) view.
2. Select one or multiple passes that you want to manage.
3. Execute the action that you wish over the selected pass or multiple passes:
 - a. Drag and drop - drag the already selected passes to the position you want to place them at.

Note

- You can move the selected passes only within the [Pass Folder](#) of the opened job. The selected passes cannot be moved to the [Pass Folder](#) of another job.
- Moving multiple passes using drag and drop groups the selected passes in their current order and moves them to the spot you have performed the drop at.

Reordering of passes in the [Job Folder](#) by drag and drop is executed depending on the direction you are moving the selected passes:

- Move up - You need to drop the selected passes directly above or on the pass you wish them to be positioned at. All selected passes will be moved before the pass you are hovering.
- Move down - You need to drop the selected passes directly after or on the pass you wish them to be positioned at. All selected passes will be moved after the pass you are hovering.

- b. Move up or move down – Click on the `Arrow Up` or `Arrow Down` buttons to move the selected passes one position up or down.

Note

Moving multiple passes using the `Arrow Up` or `Arrow Down` buttons moves every selected pass one position up or down from its current position.

- c. Choose the option you want from the [Context](#) menu - There are two ways to open the [Context](#) menu: by right-clicking anywhere in the pass folder without selecting a cell or by left-clicking a cell to select it and then using the right click of the mouse. The following options are available in the [Context](#) menu:
 - [New](#) - This option creates a new pass in the pass folder. For more information, see [Adding a Pass to a Job \[page 308\]](#).

- [Open](#) - This option opens the selected pass or multiple passes.
- [Cut](#) - This option cuts the selected passes from the [Pass folder](#).
- [Copy](#) - The option copies the selected passes from the [Pass folder](#).
- [Paste](#) - All the selected (cut or copied) passes are pasted at the end of the passes list in the [Pass Folder](#) that you have selected to paste them in.

📘 Note

- The option is available only if you have already copied or cut some passes.
- When you select to cut passes, a confirmation window for each pass appears upon the [Paste](#) execution in order to confirm that you want to delete them from their source [Pass folder](#).

- [Delete](#) - The option deletes the selected passes from the [Pass folder](#). A dialog window appears for each selected pass and prompts you to confirm its deletion.
- [Rename](#) - The [Rename](#) option enables you to enter a new name for the selected pass.
- [Move Up](#) and [Move Down](#) - This option moves the selected passes one position up or down.
- [Enable Pass](#) - This option enables all selected passes.

📘 Note

The option is visible only if the selected passes are [Disabled](#).

- [Disable Pass](#) - This option disables all selected passes.

📘 Note

The option is visible only if the selected passes are [Enabled](#).

3.1.11.12.5 Defining a Repository for a Pass

Use the [General](#) tab to select a repository for the pass.

Context

The repository variables and constants can be referenced when configuring the pass. The repository is normally the source of the From-pass and the destination of a To-pass.

To define a repository for the pass, do the following:

Procedure

1. Select the [Passes](#) tab in the job view and select the pass.

2. Choose [Open](#) from the context menu to open the pass properties and select the [General](#) tab.
3. Select the repository for the pass. The value [--None/Inherited--](#) is defined by default.

This repository is used as follows:

- If the pass belongs to a regular job, the repository selected for the pass is used.
 - If the pass belongs to a repository job, the repository selected for the repository job is used. The repository selected for the pass is not taken into account.
 - If the pass belongs to an action task that is part of a process, this field is disabled, and it is not possible to select a repository for the pass. The repository selected for the process is used.
4. Choose  [File](#)  or press `Ctrl` + `S` to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[General \(Passes\) \[page 1030\]](#)

3.1.11.12.6 Defining Custom Scripts for a Pass

Use the [General](#) tab to define custom scripts for the pass.

Context

To define custom scripts for the pass, do the following:

Procedure

1. Select the [Passes](#) tab in the job view and select the pass.
2. Choose [Open](#) from the context menu to open the pass properties and select the [General](#) tab.
3. Select a script from the list to define one or all of the following scripts:
 - Initialization script
 - Termination script
 - Entry script

Choose [Edit...](#) to open the script view and edit the script. If no script is specified, choosing [Edit...](#) you can create a new script.

4. Choose  [File](#)  or press `Ctrl` + `S` to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[General \(Passes\) \[page 1030\]](#)

3.1.11.12.7 Specifying the Source for a From ASCII File Pass

When configuring a From ASCII file pass, you use the [Source](#) tab to enter information about the source of the pass.

Context

Do the following:

Procedure

1. Select the [Passes](#) tab in the job view and select the pass.
2. Choose [Open](#) from the context menu to open the pass properties and select the [Source](#) tab.
3. Specify the details about the source file for your data.

For details about the fields, see *Source (From ASCII File Pass)*.

4. Choose  [File](#)  or press Ctrl + S to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[From ASCII File \[page 1032\]](#)

[Source \(From ASCII File Pass\) \[page 1036\]](#)

3.1.11.12.8 Specifying the Source for a From Database Pass

When configuring a From Database pass, you use the [Source](#) tab to enter information about the data source of the pass to be processed.

Context

Do the following:

Procedure

1. Select the [Passes](#) tab in the job view and select the pass.
2. Choose [Open](#) from the context menu to open the pass properties and select the [Source](#) tab.
3. Specify the connection to the source database. Use an SQL query to define the information to be read from the data source.

For details about the fields, see *Source (From Database Pass)*.

4. Choose  [File](#)  or press `Ctrl` + `S` to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[From Database](#) [page 1033]

[Source \(From Database Pass\)](#) [page 1038]

3.1.11.12.9 Specifying the Source for a From Custom Pass

When configuring a From Custom pass, you use the [Source](#) tab to enter information about the data source of the pass (a Java class) to be processed.

Context

Do the following:

Procedure

1. Select the [Passes](#) tab in the job view and select the pass.
2. Choose [Open](#) from the context menu to open the pass properties and select the [Source](#) tab.
3. Enter the name of the Java class to be called in the [Pass type](#) field. It is the base name of the class file containing the custom pass.
4. Enter the parameters that are submitted to the class when it is called. Enter the parameter names and the corresponding values in the [Parameter table](#) field.
5. Choose [File](#) [Save](#) or press `Ctrl` + `S` to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[From Custom](#) [page 1033]

[Source \(From Custom Pass\)](#) [page 1039]

3.1.11.12.10 Specifying the Source for a From LDAP Directory Pass

When configuring a From LDAP directory pass, you use the [Source](#) tab to enter information about the data source of the pass to be processed, a directory server using the LDAP protocol.

Context

Do the following:

Procedure

1. Select the [Passes](#) tab in the job view and select the pass.
2. Choose [Open](#) from the context menu to open the pass properties and select the [Source](#) tab.
3. Specify the connection to and details about the LDAP directory that the data will be read from.

For details about the fields, see *Source (From LDAP Directory Pass)*.

4. Choose [File](#) > [Save](#) or press `Ctrl` + `S` to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[From LDAP Directory](#) [page 1034]

[Source \(From LDAP Directory Pass\)](#) [page 1041]

3.1.11.12.10.1 Using Advanced Search Criteria (From LDAP Directory Pass)

Context

Note

This function is used only when searching servers that support LDAP version 2 or LDAP 3 servers that do not support paging.

The Identity Center can try to resolve the problems that arise if a pass attempts to retrieve more entries than the maximum allowed by the size or time limits set by the server.

To use the advanced search criteria, proceed as follows:

Procedure

1. Select the *Passes* tab in the job view and select a *From LDAP Directory* pass.
2. Choose *Open* from the context menu to open the pass properties and select the *Source* tab.
3. Choose the *Advanced* tab and provide the following information:

Note

To be able to use size/time limit resolving, the LDAP URL has to include one and only one asterisk (*), for instance (sn=*).

Field	Description
<i>Try to Resolve Size/Time Limit Problems</i>	Select <i>Try to resolve Size/Time Limit Problems</i> to enable the function.
<i>Characters to Process in Case of Size/Time Limit</i>	You can specify how the runtime engine should break up the character sequence in order to resolve size/time limit problems.
<i>Maximum Number of Recursive Loops</i>	Fill in the field with a number that indicates how many times the filter should be tried. This parameter is used to prevent infinite loops. In practice, the number 5 should be sufficient for most data sources, unless the data volume is very large and a great deal of data has the same 5 first characters as the search parameter.
<i>From Database</i>	Enter the name of the database where the SQL queries are stored. How you reference the database, depends on the runtime engine you are using. When using the Java runtime engine the database must be referenced using an JDBC URL. You can use the JDBC URL wizard to create the URL.
<i>From SQL Statement</i>	Enter the SQL statement you want to use to retrieve the search filter from the database. If the pass reads from the identity store, it is recommended to use one of the views installed with the product. The column names in the database query will replace any matching variables (%<ColumnName>%) in the LDAP URL.

Field	Description
	For instance, if you want to replace the host part of the URL with the result of an SQL query, you can enter an SQL query: SELECT host FROM hosts. Note When using Microsoft SQL Server, it is recommended that all SELECT queries should include with (no lock). This will reduce the table locking, improve performance, and reduce the probability of deadlocks. For instance: <pre>SELECT <columns> FROM <table> WITH (NOLOCK) WHERE <column>=<'value>'</pre> The LDAP URL may look like this: LDAP://%host%/o=myorg??SUB?(sn=*) Or if you have a table of e-mail addresses and want to retrieve the corresponding LDAP entries, you can enter an SQL query such as: SELECT mailaddr FROM mymailinglist. The LDAP URL may look like this: LDAP://host.domain.com/o=myorg??SUB?(mail=%mailaddr%) Note To avoid opening a new connection to the directory server for each value returned from the database, use the variable as part of the filter and not starting point in the LDAP URL.

4. Choose [Save](#).

Example

If "Characters to process" is set to "ABC", and the filter in the LDAP URL is set to (sn=*), the Identity Center will first try the filter (sn=A*). If a time or size limit is returned, the filter (sn=AA*) will be used. Then (sn=AB*) and (sn=AC*). The next filters will be (sn=B*), (sn=BA*), (sn=BB*), and so on.

Related Information

[Using Views to Access Identity Store Information \[page 131\]](#)

3.1.11.12.11 Specifying the Source for a From Generic Pass

When configuring a From Generic pass, you use the [Source](#) tab to enter information about the source of the pass.

Context

All data processing is handled by scripts (written e.g. in JScript or PerlScript). The scripts are run in the following sequence:

1. The data source is opened by the script specified in the [Open data source](#) field.
2. For each entry in the data source, the script specified in the [Next data entry](#) field is run and the temporary database is updated using the contents of the [Definitions table](#) in the [Destination](#) tab.
3. The data source is closed using the script specified in the [Close data source](#) field.

You define the scripts in the following way:

Procedure

1. Select the [Passes](#) tab in the job view and select the pass.
2. Choose [Open](#) from the context menu to open the pass properties and select the [Source](#) tab.
3. Specify the scripts to open, read and close the data source. View and edit the script by choosing [Edit...](#) to the right of the fields. If no script is specified, choosing [Edit...](#) you can create a new script.

For details about the fields, see [Source \(From Generic Pass\)](#).

4. Choose [File](#) > [Save](#) or press **Ctrl** + **S** to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[From Generic \[page 1035\]](#)

[Source \(From Generic Pass\) \[page 1045\]](#)

3.1.11.12.12 Specifying the Source for a From LDIF File Pass

When configuring a From LDIF file pass, you use the [Source](#) tab to enter information about the source of the pass.

Context

Do the following:

Procedure

1. Select the [Passes](#) tab in the job view and select the pass.
2. Choose [Open](#) from the context menu to open the pass properties and select the [Source](#) tab.
3. Specify the file name of the source file. Choose the ... button to the right of the field to browse the file system to find the correct file and directory. You can also specify remote files using the http and ftp protocols (http:// and ftp://).
4. Specify the character set of the LDIF file. Choose one of the following values:
 - [ANSI](#)
 - [UTF-8](#)
 - [ANSI DN/UTF-8 attr](#)
5. Choose ► [File](#) ► [Save](#) ► or press `Ctrl` + `S` to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[From LDIF File \[page 1036\]](#)

[Source \(From LDIF File Pass\) \[page 1046\]](#)

3.1.11.12.13 Specifying the Temporary Database (Destination) for From-Passes

From-passes use a temporary database as the destination for the data.

Context

To specify the information about the destination temporary database use the [Destination](#) tab of the pass properties.

Procedure

1. Select the [Passes](#) tab in the job view.
2. Select the pass and choose [Open](#) from the context menu.
3. Select the [Destination](#) tab.
4. Select either the Identity Center database or other database as the temporary database.

When using other databases than the Identity Center database, specify the JDBC connection string. Choose ... to the right of the [Database](#) field to use the JDBC connection wizard and fill in the necessary parameters.

You may also encrypt the provided connection string by selecting the option [Encrypt connection string](#).

5. In the [Table name](#) field, enter a name for the database table in the temporary database. If the same temporary database is used for several data sources, the table name should normally be different for the different data sources. Avoid using characters that are not allowed for table names.
 6. Specify how the table in the temporary database is updated in the [Table update](#) list. You can choose one of the following values:
 - [Delete table before loading](#)
The database table is deleted before the new data is loaded. The table will contain only the newest data set. This is the default value.
 - [Add entries to table](#)
New entries are appended to the temporary database when the pass is run. This must be selected if several passes write to the same table.
-  **Note**
Entries are never deleted from the table.
7. Choose  [File](#)  or press Ctrl + S to save any changes made to the pass properties.

Next Steps

Also specify the pass definitions in the [Destination](#) tab.

Related Information

[Specifying Pass Definitions for From-Passes \[page 324\]](#)

[Destination \(From-Passes\) \[page 1046\]](#)

3.1.11.12.14 Specifying Pass Definitions for From-Passes

Pass definitions for From-passes are defined in the [Destination](#) tab of the pass properties.

Context

To specify the pass definitions, do the following:

Procedure

1. Select the [Passes](#) tab in the job view, then select and open a pass (choose [Open](#) from the context menu).
2. Select the [Destination](#) tab.
3. Use the definitions table to enter the definitions for the pass. The first line is the naming attribute.

Note

You can use a template to simplify the work with the pass definitions ([Insert template](#) button).

Column name	Description
Prefix	You may choose a prefix for each definition in the first column of the definitions table. Prefixes are used to control the behavior of the following definition. For more information about the available prefixes, see Prefix Options (From-Passes) [page 1050] .
Target	Specifies the column to be created in the temporary database.

Column name	Description
	 Note You can use any legal column name (see the documentation for Microsoft Access), except <i>ID</i>, which is used by the Identity Center. If you have selected TABLE as the data type, a ... button is displayed where you can specify the sub-table. To specify the sub-table, see Defining a Sub-Table in Pass Definitions for From-Passes [page 326].
<i>Source</i>	Specifies the column name in the data source being read. You can use constants or variables when specifying the source.
<i>Script</i>	Select one of the defined scripts to be used in processing the entries.
<i>Data type</i>	Select a data type from the list: STRING With length up to 255 characters. VARCHAR NVARCHAR MEMO A MEMO field can be as long as 65535 characters. DATE Length is ignored for DATE fields. INT A 4-byte integer. TEXT TABLE Use this data type to create a sub-table to contain multiple values. Define the sub-table in the Target field.  Note The data types drop-down list is database independent. The selected type from the list is internally transformed to a proper one for the target database. The data type field accepts free text which enables you to set an option excluded from the list. Mind that the data type should be compatible with the used database.
<i>Size</i>	Specifies the field length in the temporary database. Each data type has a default length that is inserted when you select a data type. If you have selected STRING as the data type, you can modify this value.

- Choose  **File**  or press **Ctrl** + **S** to save any changes made to the pass properties.

Related Information

[Destination \(From-Passes\) \[page 1046\]](#)

[Specifying the Temporary Database \(Destination\) for From-Passes \[page 323\]](#)

3.1.11.12.14.1 Defining a Sub-Table in Pass Definitions for From-Passes

When entering pass definitions (in the [Destination](#) tab) for a From-pass, you can create a sub-table in the temporary database if you want to split a multivalue field into separate values.

Context

This enables a "reverse lookup" to find all entries with a specific value. The unique identifier (IID) in the main table is used as a reference (RefID) in the sub-table.

You create a sub-table by selecting [TABLE](#) as the data type for a column in the temporary database. You use the [Define sub-table](#) dialog box to specify the properties of the sub-table.

To define or modify the sub-table, do the following:

Procedure

1. Use the ... button in the [Target](#) column in the pass definitions.

This will open a [Define sub-table](#) dialog box. Specify the properties of the sub-table:

Table name	Enter the name of the sub-table. The name of the target column is suggested, but you can modify this value.
Column name	Enter the name of the column in the table specified in the Table name field of the Destination tab.
Data type	Select a data type from the list. You can choose between the following values: • STRING With length up to 255 characters. • VARCHAR With length up to 255 characters. • NVARCHAR With length up to 255 characters.

	• MEMO A MEMO field can be as long as 65535 characters. • DATE Length is ignored for DATE fields. • INT A 4-byte integer. • TEXT
Field size	Enter the size of the column. The default value is 255 characters. Size is only relevant if you have selected STRING in the Data type field.
Multivalue separator	Enter the character that separates values in fields containing multiple values in the source file. If the separator is space or tab, enter the words SPACE or TAB (without quotes).

2. Choose **OK** to close the dialog box, then continue specifying the pass definitions.

3.1.11.12.15 Specifying the Source for To-Passes

When you specify the source for the passes, you use the **Source** tab in pass properties. The **Source** tab for all To-passes have the same fields.

Context

Do the following:

Procedure

1. For a job, select the **Passes** tab in the job view and select a pass.
2. Choose **Open** from the context menu to open the pass properties and select the **Source** tab.
3. Select one of the following as the data source for the pass:

Option	Description
Identity Store	Select the correct identity store from the list. If identity store is selected as the source, the options for both defining a filter and a SQL query (the fields Filter and SQL Query are enabled. Fields Database and Encrypt Connection String are disabled.

Option	Description
Identity Center Database	Fields Database and Encrypt Connection String , including the possibility to use a filter, are disabled if Identity Center database is selected as the source. You are still able to define a SQL query.
Database	Select this option for other databases than the Identity Center database as the source. When using other databases than the Identity Center database, specify the JDBC connection string. Choose ... to the right of the Database field to use the JDBC connection wizard and fill in the necessary parameters. You may then also encrypt the provided connection string by selecting the option Encrypt Connection String. The possibility to use a filter is disabled if Database is selected as the source. You are still able to define a SQL query.

4. Define the data to be read from the source. You can choose between the following options:

Option	Description
Filter	This field is only available if Identity Store is selected as the source. Select the option and then enter the filter in the field that appears. You can use the filter editor to create the filter (the Edit... button just above the right upper corner of the field). The filter will automatically generate SQL query. To view the query, select the SQL Query option. For details on how to use the filter editor, see <i>Using the Filter Editor</i>. You can also choose to include inactive entries or use attribute IDs instead of attribute names (options Include Inactive Entries and Use Attribute ID Instead of Attribute Name in SQL Query respectively). Note If you select the SQL Query option and then manually edit the query, the Filter option will be disabled. To activate Filter again, you must clear the SQL query field.
SQL Query	Select the option and then enter the SQL in the field that appears. You may use the Filter option to help you generate an SQL query, that you so can extend. Note If you manually edit the SQL query, the Filter option will be disabled. To activate Filter again, you must clear the SQL query field. When the values are retrieved (read) from the identity store, it is recommended to use one of the views installed with the product. Note The SQL query should not be terminated or delimited with semicolon.

Option	Description
	Note When using Microsoft SQL Server, it is recommended that all SELECT queries should include "with (nolock)". This will reduce the table locking, improve performance, and reduce the probability of deadlocks. For instance: <pre>SELECT <columns> FROM <table> WITH (NOLOCK) WHERE <column>=<value></pre> If you specified an XML recordset file in the Database field, you only specify the WHERE clause, for instance: <pre>(cn like 'a%')</pre> This will then be used as a filter for processing the XML recordset file.

Depending on the data source, you may need to prepare it in different ways. For instance, to read data from Excel, you need to name an area in the spreadsheet to be able to refer to it.

5. Choose **File > Save** or press **Ctrl + S** to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[Source \(To-Passes\) \[page 1057\]](#)
[Using the Filter Editor \[page 269\]](#)

3.1.11.12.16 Specifying the Destination for a To Identity Store Pass

When configuring a To Identity store pass, you use the [Destination](#) tab to specify information about the identity store you are going to update.

Context

Entry types (see [Entry Types](#)) provide rules for entries written to the identity store. The entry type contains information about permitted and mandatory attributes, and this list of attributes will be used during updates of the identity store. If the job creates entries in the identity store, the entries will be of this entry type.

If you have selected [Automatically create attributes](#) for the identity store, any new attributes that are added to the entries written with the job will be added to the entry type as well.

If a mandatory attribute is missing, the whole operation is stopped and all changes are rolled back. An error message is written to the job log. The error message contains the name of the first mandatory attribute that is missing. The same applies if the job tries to remove a mandatory attribute.

If you add an entry type to an entry that did not have an entry type previously, the operation will fail and be rolled back if any of the mandatory attributes is missing.

To specify the destination, proceed as follows:

Procedure

1. Select the [Passes](#) tab in the job view and select the pass.
2. Choose [Open](#) from the context menu to open the pass properties and select the [Destination](#) tab.
3. Specify the information about the destination identity store.

For details about the fields, see [Destination \(To Identity Store Pass\)](#).

4. Use the [Definitions table](#) to enter definitions for the pass. The first line is the naming attribute.

Press `Ctrl` + `Space` to look up for the available options or to activate autocompletion.

Note

You can use a template to simplify the work with the pass definitions ([Insert Template](#) button). For more information, see [Using a Template \(To Identity Store Pass\)](#).

Note

All variables and constants are enclosed in a % pair (%name%). If there is a % sign in the output, you must use the keyword \$PERCENT.

Option	Description
Prefix	Select one of the prefixes to control the behavior of the following definition. For more information about the available prefixes, see Prefix Options (To-Passes) [page 1074]
Attribute	Specify the column in the database.
	Note Not all characters are allowed in attribute names. We recommend using the following characters: A-Z/a-z, 0-9 and _.
Value	Specifies the source for the value, or how the value is to be generated.
	Note Make sure you are familiar with the usage of attribute operators, as described in Using Attribute Operators (To Identity Store Pass) [page 337]

5. Choose [File](#) > [Save](#) or press `Ctrl` + `S` to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[Destination \(To Identity Store Pass\) \[page 1059\]](#)

[Using a Template \(To Identity Store Pass\) \[page 331\]](#)

[Entry Types \[page 71\]](#)

3.1.11.12.16.1 Using a Template (To Identity Store Pass)

You can use a template to simplify work with definitions.

Prerequisites

- You have configured the Java system properties of your SAP NetWeaver Application Server Java for the [idmdevstudio](#) application. See the Related Information section.

Context

Choose between the following methods to create the template:

- [Data source template](#)
- [Data destination template](#)

Procedure

1. Select the [Passes](#) tab in the job view, then select and open a pass (choose [Open](#) from the context menu).
2. Select the [Destination](#) tab.
3. Choose [Insert template](#) above the definitions table and select one of the following:

Option	Description
Data source template	To insert a template based on the data source. The Identity Center tries to identify as much as possible of your data source and creates a template. Each column in the data source corresponds to one line in the template.
Data destination template	To insert a template based in the target identity store.

Option	Description
	The Identity Center analyzes the identity store and inserts the column names in the Attribute column. Use the context menu to fill in the corresponding columns from the data source in the Value column.

4. Choose **File > Save** or press **Ctrl + S** to save any changes made to the pass properties.

Related Information

[Configuring the Java System Properties](#)

3.1.11.12.16.2 Specifying Properties When Writing Attribute Values (To Identity Store Pass)

When specifying the values of attributes in a To identity store pass, you can use a specific syntax to specify properties of a value. This can for instance be used to specify a validity period for a value or a context.

There are different properties depending on the attribute type. Structural attributes and other reference attributes are handled in the same way as regular attributes in the following description. Assignments are described separately.

The generic syntax for specifying attribute properties is as follows:

```
<AttributeName> {<PropertyName>=<PropertyValue>!!  
<PropertyName>=<PropertyValue>!!...}<Value>
```

❖ Example

```
MXREF_MX_ROLE {VALIDFROM=2010-06-01!!VALIDTO=2010-12-31}<Manager>|  
{VALIDFROM=2011-01-01}<Developer>  
MXREF_MX_ROLE {CTX=<Trondheim>}<Store Manager>
```

<AttributeName>

This can be any attribute in the identity store.

<PropertyName>

This can either be one of the operators or properties.

Operators for assignments

The operators are used to indicate a given behavior and does not correspond to a given property.

Operator	Description
DIRECT_REFERENCE=1	This operator is used to disable any Validate or Member event process when assigning/deassigning a role from a job. This is the same as you can specify for each role or privilege with the attributes MX_ADDMEM_DISABLE_POLICY and MX_DELMEM_DISABLE_POLICY described in Identity Store Schema document.
BYPASS_VALIDATE_TASK=1	This operator is used to disable a Validate add, Validate remove or Validate modify validity process when assigning/deassigning a role or privilege. The assignment/deassignment is automatically approved.
BYPASS_MEMBER_TASK=1	This operator is used to disable an Add member, Remove member or Modify validity process when assigning/deassigning a role or privilege. The event processes are not executed when the assignment/deassignment is performed.
BYPASS_MODIFY_TASK=1	This operator is used to disable a Modify process when assigning/deassigning a privilege or modifying an attribute, which is part of trigger attributes for a system.

Note

The properties above will be considered only for direct assignments.

Properties for assignments

Property	Description
CTX	Reference to a context for this assignment. Note This property can only be used in the following cases: - When creating a reference attribute, not when modifying it. - When the reference attribute's mxi_link.mclink-Type = 2. This means that you have privilege to person, or role to person type of assignment, as the context-based assignments always involve assigning a role or a privilege with a context to the person.
LinkAssigner	MSKEY of the user adding the assignment.

Property	Description
LinkReason	Reason for this assignment.
ProcessInfo	Info describing the modification of the attribute. The syntax is the same as for UserId.
Repository	A reference to the repository ID that should be defined as owner of this attribute.
RequestID	The Request ID for the assignment.
ValidFrom/ValidTo	Validity period of the assignment.

Properties for regular attributes

Property	Description
Repository	A reference to the repository that should be defined as owner of this attribute.
UserId	The UserId info for the modification of the attribute.
ValidFrom/ValidTo	Validity period of the assignment.

<PropertyValue>

A legal value of the given property.

Legal characters - The following characters are reserved and cannot be used in a property value:

- . Period
- [] Square brackets
- <> Brackets
- {} Curly brackets
- !! Two exclamation marks

Dates and times - Dates and times should always be written on ISO8601 format: **YYYY-MM-DD** or **YYYY-MM-DDThh:mm:ss**

<Value>

This is the attribute value.

❖ Example

Specifying Attribute Validity

To assign the role Manager to a user for two separate periods, a To Identity store pass can write the following value:

```
MXREF_MX_ROLE {VALIDFROM=2010-04-01!!VALIDTO=2010-04-30}<Manager>|  
{VALIDFROM=2010-06-01!!VALIDTO=2010-06-30}<Manager>
```

You can also use this syntax on the MSKEYVALUE attribute. In this case, the whole entry will be removed at the specified date. As MSKEYVALUE normally is the naming attribute, the validity period must be specified on a separate line:

```
MSKEYVALUE %EmployeeID%  
MSKEYVALUE {VALIDTO=2009-12-31}%EmployeeID%
```

Note

In general, it is not recommended to remove entries from the identity store.

❖ Example

Bypass Modify Task

You can use BYPASS_MODIFY_TASK to skip the triggering of the modification plugin of a system. This operator could be used in combination with other modifications, for instance add/remove of assignments or add/modify/delete of trigger attributes.

Let's assume that the attributes MX_FIRSTNAME and MX_LASTNAME are part of the trigger attributes of some Java system privilege. If you want to modify the first name of a user, without triggering the modification process to the related Java system, you should use the BYPASS_MODIFY_TASK operator:

```
MSKEYVALUE - someJavaUser  
changeType - modify  
MX_FIRSTNAME - NewFirstName  
MXREF_MX_PRIVILEGE - {BYPASS_MODIFY_TASK=1}<PRIV:SYSTEM:JAVA>  
MX_LASTNAME - NewLastName  
MXREF_MX_PRIVILEGE - {BYPASS_MODIFY_TASK=1}<PRIV:SYSTEM:JAVA>
```

SAP Identity Management will detect that the first and the last name of a user are changed, but will not trigger the modification event to update the Java system, because BYPASS_MODIFY_TASK operator is added after the attribute modification.

Note

The operator should be added **after each** of the system trigger attributes, which usually trigger the modification process.

❖ Example

Bypassing Member Event Processes

You can use the BYPASS-property to specify that the member event process is not to be executed. You need to use this property before any privilege or role for which you don't want the member event process to be executed.

This can for instance be used to clean up situations where a Remove member process has failed, and the role/privilege is still there. You could do this with the following statements in a To identity store pass to remove the privilege MyPrivilege without executing the Validate remove and Remove member processes:

```
{D}{BYPASS_VALIDATE_TASK=1} <MyPrivilege>
{D}{BYPASS_MEMBER_TASK=1} <MyPrivilege>
```

You can also use the To DIRECT_REFERENCE property for instance to disable the Add member process when assigning the role "Manager" to a user:

```
MXREF_MX_ROLE {DIRECT_REFERENCE=1}<Manager>
```

❖ Example

Specifying Context

To specify a given context for an assignment, you can either reference the MSKEY of the context entry, or the MSKEYVALUE, enclosed in <>.

```
MXREF_MX_ROLE {CTX=42}<Manager>
MXREF_MX_ROLE {CTX=<Trondheim>}<Store Manager>
```

3.1.11.12.16.3 Removing Attributes (To Identity Store Pass)

You can remove an attribute by writing "%%" which will work in Java runtime engine. If this is a multivalue attribute, all values are removed. Any delete event process defined for the attribute will be started.

Context

To remove an attribute, proceed as follows:

Procedure

1. Select the [Passes](#) tab in the job view, then select and open a pass (choose [Open](#) from the context menu).
2. Select the [Destination](#) tab.
3. In the [Definitions table](#), enter the value [%%](#) for the attribute you want to remove.

Using the {R} operator with roles and privileges

When using the {R} operator, all privileges (or roles) will first be removed, and the corresponding Remove member processes are executed. Immediately afterwards, they are added, and the Add member processes are executed. If this is not the behavior you intend, you should use the {M} operator, which only executes the delta.

There is a potential problem with the Remove member processes started by using the {R} operator. If any of these processes are retried, or the execution is delayed for some other reason, the Add member process may be executed before the corresponding Remove member process. Depending on what the processes do, this can cause the resources to be de-provisioned.

3.1.11.12.16.4 Using Attribute Operators (To Identity Store Pass)

You can use a number of operators to control how attribute values are written to the identity store.

There are different operators for multivalue and single-value attributes. The following are valid attribute operators which can be used in a To Identity Store pass: {A}, {D}, {d}, {e}, {R}, {E}, {M} and {T}.

For more information about the usage of attribute operators, see the relevant topics below.

Attribute Operators Syntax

The following syntax rules apply for attribute operators:

- {R}, {E}, {M} and {T} operators should always be placed at the beginning of the corresponding attribute value.
These operators cannot be combined, neither with each other, nor with other operators in a multivalue attribute. The only exception from this rule is the default {A} operator, which can be combined with all attribute operators. The reason for this is that these operators can perform more than one type of operation on a given entry and attribute, for example: to delete and add values. Placing one more operator could lead to unexpected provisioning and inconsistent data.

❖ Example	
Value	Result
{R}CurrentDepartment {M}NewDepartment	Wrong syntax, you cannot combine these operators.
{M}CurrentOrganization {A}NewOrganization	Correct syntax. However, in this case it is sufficient to only provide the value you want to add after the {M} operator, that is {M}CurrentOrganization

- {D}, {d}, {e} and {A} operators can be combined with each other in a multivalue attribute. It is not mandatory for them to be placed at the beginning of the corresponding attribute value.
If some value in the multivalue string doesn't have an attribute operator defined, SAP Identity Management will automatically append the first operator present in the values on the left side. If there is no operator defined for the values on the left side, then the default {A} operator will be used.

❖ Example

Value	Result
Department4 Department5 {D}Department1 Department2 Department3 {e}FutureDepartment1 FutureDepartment2 {A}FutureDepartment3	This will be implicitly transformed into the following multivalue string: <pre>Department4 Department5 {D}Department1 {D}Department2 {D}Department3 {e}FutureDepartment1 {e}FutureDepartment2 {A}FutureDepartment3</pre> The empty values in this example: will not receive an operator and will be skipped later.

For attributes of type entry reference, you can provide the referenced entry's ID in the following ways:

- Write the entry's MSKEY. For example: **{A}12345** for MXREF_MX_PRIVILEGE
- Write the entry's MSKEYVALUE in **<>** angle brackets. For example: **{A}<Smith>** for MX_MANAGER

Operators for Multivalue Attributes

Some attributes in the identity store are defined as multivalue attributes. When configuring the destination of a To Identity Store pass, there are a number of operators you can use to specify how to handle updates to such attributes:

- [Add Operations with {A} Operator \[page 341\]](#)
- [Delete Operations with {d}, {D} and {e} Operators \[page 342\]](#)
- [Replace Operations with {R} and {E} Operators \[page 345\]](#)
- [Merge Operations with {M} and {T} Operators \[page 347\]](#)

Event Handling

If member event handling is defined, updating an attribute in this way may trigger the corresponding event process.

❖ Example

In the following examples it is assumed that this is a privilege reference (MXREF_MX_PRIVILEGE) and that all privileges have add/remove member process. The value within brackets {} are properties of the value, such as ValidTo, Reason and Repository. This is the current value that is present in the Identity Management database for the given entry:

A

{Repository=5}B

{ValidTo=<tomorrow>}C

Definition	Value after update	Event tasks
A B D	A {Repository=5}B {ValidTo=<tomorrow>}C D	Add D
{R}A B D	A B D	Remove A, B, C Add A, B, D
{M}A B D	A {Repository=5}B D	Remove C Add D
{M}A {LINKID=<LinkID>!!ValidTo=<nextweek>}B D	A {Repository=3!!LINKID=<LinkID>!!ValidTo=<nextweek>}B D	Remove C Add D
{M}B {LINKID=<LinkID>!!ValidTo=<nextweek>}C D	{Repository=5}B {LINKID=<LinkID>!!ValidTo=<nextweek>}C D	Remove A Add C, D

Note

B is not executed as the time period is reduced (from indefinite to <nextweek>).

Note

C is executed because the time period is extended.

Which of the event processes are executed, depends on how the member event handling is configured.

Operator for Single-Value Attributes

For single-value attributes, there is one valid operator: {E}. However, depending on whether the single-value attribute is an entry reference or a regular (non-reference) attribute, the attribute values are handled differently.

Single-Value Non-Reference Attributes

For single-value non-reference attributes, the {E} operator removes all future values of the attribute but leaves the current value. No event processes are executed. You can expect the following behavior:

- If you use the {E} operator alone on a line, this will remove the value, as an empty value will be written after the {E} operator is executed.
- If only future values are added after the {E} operator, the current value will be kept. If you also want to remove this value, add an empty value at the beginning, by inserting a | (pipe) symbol before the first future value.
- If you use the {E}<VALUE1> syntax, all existing values (current and future ones) will be removed, and <VALUE1> will be set as the current value.
- If you specify more than one validity periods for an attribute value, for instance: {E}{Validity1}<VALUE1>|{Validity2}<VALUE2>, the order of the validity is very important. It should be: {E}{Validity1= current validity period}<VALUE1>|{Validity2= future validity period}<VALUE2>. In this case, the {E} operator will set VALUE1 as the current value and it will also create a PVO for VALUE2 to set it as the next attribute value.
For example: MX_LASTNAME {E}{Validity1= current validity period}<Anderson>|{Validity2= future validity period}<Johnson>
- If you define the validity period in wrong order, for instance {E}{Validity1=current validity period}<VALUE1>|{Validity2= past validity period}<VALUE2>, the {E} operator will try to set VALUE2 as the next attribute value. Since its validity period is in the past, the operator will set VALUE2 as a history value of the attribute which will result in empty current value of the attribute.

❖ Example

Definition	Current value after update	Future value(s) after update
{E}	<Removed>	<Removed>
{E}XYZ	XYZ	<Removed>
{E}NULLATTR	<Unchanged>	<Removed>
{E}{ValidFrom=<date>}X	<Unchanged>	X
{E} {ValidFrom=<date>}X	<Removed>	X
{E}XYZ {ValidFrom=<date>}X	XYZ	X

Single-Value Entry Reference Attributes

For single-value entry reference attributes, the {E} operator removes all existing values: current and future ones. You can expect the following behavior:

- If you specify more than one attribute values, for instance: {E}{Validity1}<ENTRY_1>|{Validity2}<ENTRY_2>, the order is very important, because the {E} operator will handle the values as follows: first, the <ENTRY_1> will be created with its validity and then <ENTRY_2> will be created.
- If you provide attribute value without validity, for instance: {E}<ENTRY_1>, in this case, <ENTRY_1> will be created as active (LinkState=0) link without validity (ValidFrom = null and ValidTo = null).

- If you specify a future attribute value, for instance: {VALIDFROM=2051-01-01!!VALIDTO=2100-12-31}<ENTRY_1>, it will be created as inactive (LinkState=1) link in the link table and no PVO will be created. The link will be activated when its ValidFrom < current_date> is reached.

In the following examples, it is assumed that the modified attribute is a single-value entry reference attribute.

❖ Example

Definition	Current value after update	Future value(s) after update
{E}	<Removed>	<Removed>
{E}{VALIDFROM=2000-01-01!!VALIDTO=2050-12-31}<ENTRY_1>	<ENTRY_1>	<Removed>
{E}{VALIDFROM=2051-01-01!!VALIDTO=2100-12-31}<ENTRY_1>	<Removed>	<ENTRY_1>
{E}{VALIDFROM=2000-01-01!!VALIDTO=2050-12-31}<ENTRY_1> {VALIDFROM=2051-01-01!!VALIDTO=2100-12-31}<ENTRY_2>	<ENTRY_1>	<ENTRY_2>
{E}<ENTRY_1>	<ENTRY_1>	<Removed>
{E}<ENTRY_1> <ENTRY_2>	<ENTRY_2>	<Removed>
{E}<ENTRY_1> {VALIDFROM=2051-01-01!!VALIDTO=2100-12-31}<ENTRY_2>	<ENTRY_1>	<ENTRY_2>

Related Information

[Removing Attributes \(To Identity Store Pass\) \[page 336\]](#)

[Modifying Properties of Assignments \[page 354\]](#)

3.1.11.12.16.4.1 Add Operations with {A} Operator

The {A} operator adds a given attribute value to the given entry. The {A} operator can be used with all types of attributes.

If no operator is defined for the given attribute value, then the {A} operator is used by default for add and modify operations.

For attributes of type entry reference, you can provide the referenced entry's ID in the following ways:

- Write the entry's MSKEY
- Write the entry's MSKEYVALUE in **<>** angle brackets

Note

When writing new attribute values, avoid any whitespaces between **<>** (angle brackets), **{ }** (curly brackets), **=** (equals signs), **!!** (double exclamation marks) and the corresponding values.

For example, the value **{A} { ValidTo = 2040-12-01}Smith** will cause a syntax error, while the value **{A}{ValidTo=2040-12-01}Smith** will succeed.

Example

Attribute	Value	Result
MX_FIRSTNAME	{A}	No modification is triggered.
MX_MANAGER	<Smith>	Since no operator is defined, {A} is used by default for add or modify operations. In this case, it is the same as providing {A}<Smith> .
MX_LASTNAME	{A} {ValidTo=2030-12-31}Smith	Writes the value Smith of the MX_LASTNAME attribute with validity until December 31, 2030.
MXREF_MX_PRIVILEGE	{A} {BYPASS_MEMBER_TASK=1!! ValidTo=2040-06-30!! LinkReason=PositionChanges}<PRIV:ROLE:JAVA:WSDL_Viewer>	Skips the Add Member task defined on the PRIV:ROLE:JAVA:WSDL_Viewer privilege and adds the PRIV:ROLE:JAVA:WSDL_Viewer privilege only in the Identity Center with validity until June 6, 2040 and reason for adding the privilege - PositionChanges.

3.1.11.12.16.4.2 Delete Operations with {d}, {D} and {e} Operators

For delete operations, you can use the {d}, {D} and {e} operators.

Note

These operators can be combined with each other in the value string of a multivalue attribute. It is not mandatory for them to be placed at the beginning of the corresponding attribute value. For more

information, see *Attribute Operators Syntax* in [Using Attribute Operators \(To Identity Store Pass\)](#) [page 337].

Note

The attribute value provided after {d} and {e} operators is case sensitive.

{d} Operator

The {d} operator deletes the given attribute value from the given entry. If no value is given, all values are removed.

The value provided after {d} is case sensitive. That is, if the value **Smith** exists in the Identity Management database, to delete this value with {d}, you need to provide **{d}Smith** in the To Identity Store pass. **{d}SMITH** and **{d}smith** will not delete it.

The {d} operator can delete only successfully assigned reference attribute values. To delete a rejected or failed assignment, you need to use the {e} operator.

The {d} operator cannot delete future reference attribute values, matching the given value (if any). If a validity period is defined for an attribute value (that is, ValidFrom in the past or ValidTo in the future are set), the {d} operator can delete the attribute value only if you provide the corresponding Link ID (mcUniqueld column from mxi_link database table) before the attribute value in the following format: **{LINKID=12345}**.

Example

Attribute	Value	Result
MXREF_MX_PRIVILEGE	{d}	Deletes all privileges from the given entry regardless of the validity period.
MXREF_MX_PRIVILEGE	{d} {LINKID=12345}<PRIV: JAVA: ONLY>	Deletes PRIV:JAVA:ONLY, even if a current validity period is set for this privilege.
MXREF_MX_PRIVILEGE	{d}<PRIV: JAVA: ONLY>	Deletes PRIV:JAVA:ONLY, if no current validity period is set for this privilege.
MX_LASTNAME	{d}Smith	Deletes the non-reference value Smith , even if an expiry time is set.

{D} Operator

The {D} operator works the same way as {d} operator. The only difference is that {D} is case insensitive. That is, **{D}SMITH** and **{D}Smith** deletes the value **Smith**.

{e} Operator

This operator works the same way as {d} operator, but it can also do the following:

- Delete all future attribute values (any pending values with a future ValidFrom) matching the given value.

Note

For attributes of type entry reference, if the attribute value has some validity period defined (for example, ValidFrom set in the past or ValidTo set in the future), you can directly delete the value with {e} operator.

- Delete rejected or failed assignments.

Example

In this example, you have a privilege PRIV:JAVA:ONLY assigned to the user John Smith with ValidTo=2030-06-30. The user also has the same privilege PRIV:JAVA:ONLY assigned with ValidFrom=2031-01-01 and ValidTo=2040-06-30.

The following will happen if you use {e} and {d} operators:

Attribute	Value	Result
MXREF_MX_PRIVILEGE	{e}	Deletes all privileges assigned to the user John Smith, regardless of their validity period or whether they are successfully assigned or not.
MXREF_MX_PRIVILEGE	{e}<PRIV: JAVA: ONLY>	Deletes the current and the future record, regardless of whether they are assigned successfully or not.
MXREF_MX_PRIVILEGE	{d}<PRIV: JAVA: ONLY>	Does not delete the current record and leaves the future one unmodified.

Note

If a validity period is defined for an attribute value (that is, ValidFrom or ValidTo are set), the {d} operator can delete the attribute value only if you provide the corresponding Link ID (mcUniqueld column from mxi_link database table) before the attribute value in the following format: **{LINKID=12345}**.

Attribute	Value	Result
MXREF_MX_PRIVILEGE	<code>{d} {LINKID=123456}<PRIV: JA VA: ONLY></code>	Deletes the current record and leaves the future one unmodified.

❖ Example

In this example, the operations are done in sequence, so that the attribute value after one operation will be the input value for the next.

{R}abc | cde - This will remove all existing values and replace them with 'abc' and 'cde'.

{d}cDe - This will not do anything as there is no value matching 'cDe'.

{D}cDe - This will remove the value 'cde'.

{M}abc - No change, as the new value is the same as the existing.

{R}abc - Replaces the existing value abc with the new value abc.

3.1.11.12.16.4.3 Replace Operations with {R} and {E} Operators

For replace operations, you can use the {R} and {E} operators.

📌 Note

These operators should always be placed at the beginning of the corresponding attribute value. They cannot be combined, neither with each other, nor with other operators in a multivalue attribute. For more information, see *Attribute Operators Syntax* in [Using Attribute Operators \(To Identity Store Pass\) \[page 337\]](#).

{R} Operator

The {R} operator replaces the current values of the given attribute with the new values. This operator first deletes all current values of the given attribute and then adds the new attribute values. Any corresponding event tasks are executed before the new values are added. See [Removing Attributes \(To Identity Store Pass\) \[page 336\]](#) for details.

📌 Note

If the deletion of the current attribute values from the given entry is delayed for some reason, then the given new attribute value might be added before the deletion of the current values. This means that, the Add Member process might be executed before the corresponding Remove Member process. Depending on the process configuration and business data, this can cause unexpected deletion of attribute values. To

avoid possible issues, see the options provided in the [Merge Operations with {M} and {T} Operators \[page 347\]](#).

The {R} operator cannot replace future attribute values, that is, any pending values (values with a future ValidFrom) are left unmodified. But the {R} operator can add future attribute values. The {R} operator can replace current reference attribute values with validity period set.

❖ Example

In this example, you have a user with attribute MX_DEPARTMENT currently set to **Marketing** and also a future value set to **Operations** with validity from December 9, 2030. The following will happen if you use {R} operator:

Attribute	Value	Result
MX_DEPARTMENT	<code>{R}Finance {ValidFrom=2040-06-01}Sales</code>	Replaces the current value Marketing with the new value Finance , leaves the future value Operations unmodified and adds new future value Sales with validity from June 1, 2040.

{E} Operator

This operator works the same way as the {R} operator, but it can also replace future attribute values.

❖ Example

In this example, you have a user with attribute MX_DEPARTMENT currently set to **Marketing** and a future value set to **Operations** with validity from December 9, 2030. The following will happen if you use {R} and {E} operators:

Attribute	Value	Result
MX_DEPARTMENT	<code>{R}Finance</code>	Deletes the current value Marketing , leaves the future value unmodified and sets the new value to Finance .
MX_DEPARTMENT	<code>{E}Human Resource</code>	Deletes the current value Marketing and the future value Operations . Sets the new value to Human Resource .

Attribute	Value	Result
MX_DEPARTMENT	{E}Customer Service {ValidFrom=2027-06-30}Sales	Deletes the current value Marketing and the future value Operations . Sets the new value to Customer Service that will be changed to Sales on June 30, 2027.

For more information about how the {E} operator groups multivalue attribute values by their validity, see [3024298](#) .

3.1.11.12.16.4.4 Merge Operations with {M} and {T} Operators

For merge operations, you can use the {M} and {T} operators.

Note

These operators should always be placed at the beginning of the corresponding attribute value. They cannot be combined, neither with each other, nor with other operators in a multivalue attribute. For more information, see *Attribute Operators Syntax* in [Using Attribute Operators \(To Identity Store Pass\) \[page 337\]](#).

{M} Operator

The {M} operator merges attribute values. The list of attribute values provided after the {M} operator is compared to the list of attribute values present in the Identity Management database for the given entry.

Note

This calculation is only done for the direct assignments specified in the pass, not any inherited privilege or role assignments resulting from this assignment.

Note

Attribute values with ValidFrom set in the future cannot be deleted or changed by using {M} operator, but you can add new values with ValidFrom in the future.

- To remove a value, it should not be present after the {M} operator, but it should be present in the Identity Management database for the given entry.
- To add new value, it should be present after the {M} operator, and it should not be present in the Identity Management database for the given entry. You can do this, as described in [Specifying Properties When Writing Attribute Values \(To Identity Store Pass\) \[page 332\]](#)

- To modify a value, it should be present after the {M} operator, and it should also be present in the Identity Management database for the given entry. It is only possible for reference attribute values. You can do this as described in [Modifying Properties of Assignments \[page 354\]](#)

Note

If you are changing a property (Validity or Reason) of an assignment, add the LINKID of the assignment before the property.

- To keep a value unchanged, it should be present after the {M} operator, and it should also be present in the Identity Management database for the given entry. If the attribute is of type reference and the value is context-based, it is mandatory to provide the corresponding context MSKEY or MSKEYVALUE before the attribute value in one of the following ways: {CTX=<MSKEYVALUE>}<Value> or {CTX=MSKEY}<Value>

Note

Providing the specific context of the reference attribute value is necessary, because it is possible for one entry to have the same attribute value assigned several times with different contexts for the same attribute. Therefore, there should be a way to differentiate the distinct values other than using their names.

If you do not provide the specific context in the To Identity Store pass when you want to keep the value of a context-based reference attribute unchanged, this will result in deleting of the value.

- To change the ValidTo of a non-reference attribute value with current validity, the new expiry time and the attribute value should be present after the {M} operator and it should also be present in the Identity Management database for the given entry. To do this, add the new expiry time in the following way:
{ValidTo=<new date or datetime>}ATTRIBUTE_VALUE
For more information, see [3028156](#)

Example

In this example, we modify the MXREF_MX_PRIVILEGE multivalue entry reference attribute in the To Identity Store pass.

Values in Identity Management database	Values in To Identity Store Pass	Result
<pre><PRIV: JAVA: ONLY>, <PRIV: SYSTEM: JAVA>, <PRIV: ROLE: JAVA: SAP_SLD _GUEST>, {ValidFrom=2030-12-01 12:56:07.345!! ValidTo=2032-12-01 23:13:01.567}<PRIV: ROLE : JAVA: SAP_SDIC_SUPER_AD MINISTRATOR></pre>	<pre>{M}<PRIV: JAVA: ONLY> <PRIV: SYSTEM: JAVA> <PRIV: ROLE: JAVA: NWA_JAV A_SUPPORT></pre>	Deletes <PRIV: ROLE: JAVA: SAP_SLD_GUEST> Adds <PRIV: ROLE: JAVA: NWA_JAV A_SUPPORT> Keeps the future value of <PRIV: ROLE: JAVA: SAP_SDIC_SUPER_ADMINISTRATOR> unmodified.

Example

In this example, we modify the OFFICE_ADDITIONAL custom non-reference multivalue attribute in the To Identity Store pass.

Values in Identity Management data-base

Values in To Identity Store Pass

Result

ATTRIBUTE_VALUE_1
ATTRIBUTE_VALUE_2
Pending Value Object 1
with attributes:
MX_ATTRIBUTE_NAME =
"OFFICE_ADDITIONAL",
MX_ATTRIBUTE_VALUE =
FUTURE_ATTRIBUTE_VALUE_
1,
MX_VALIDFROM =
2030-07-01T12:30:01,
MX_VALIDTO =
2031-07-02T12:30:00

{M} ATTRIBUTE_VALUE_1
| ATTRIBUTE_VALUE_3

Deletes
ATTRIBUTE_VALUE_2
Adds ATTRIBUTE_VALUE_3
Keeps the PVO holding
FUTURE_ATTRIBUTE_VALUE_
1 and the value of
ATTRIBUTE_VALUE_1

❖ Example

In this example, we modify the SUPERVISOR_ADDITIONAL custom reference multivalue attribute in the To Identity Store pass. A context is provided for the reference attribute value.

Values in Identity Management data-base

Values in To Identity Store Pass

Result

ATTRIBUTE_VALUE_1
ATTRIBUTE_VALUE_1
assigned with context
"GERMANY"

{M}
{CTX=<GERMANY>}ATTRIBUT
E_VALUE_1 |
{CTX=<ITALY>}ATTRIBUTE_
VALUE_1 |
ATTRIBUTE_VALUE_2

Deletes
ATTRIBUTE_VALUE_1
Adds ATTRIBUTE_VALUE_1
with context "ITALY"
Adds ATTRIBUTE_VALUE_2
Keeps the value
of ATTRIBUTE_VALUE_1
assigned with context
"GERMANY"

❖ Example

In these examples, we modify the MXREF_MX_PRIVILEGE multivalue entry reference attribute in the To Identity Store pass. A context is provided for the entry reference attribute value.

Values in Identity Management database

Values in To Identity Store Pass

Result

```
<PRIV: JAVA: ONLY>,
<PRIV: SYSTEM: JAVA>,
{CTX=<BULGARIA>}<PRIV: ROLE: JAVA: SAP_SLD_GUEST>
,
{CTX=<GERMANY>}<PRIV: ROLE: JAVA: SAP_SLD_GUEST>,
{CTX=<CHINA>}<PRIV: ROLE: JAVA: SAP_SLD_GUEST>,
<PRIV: ROLE: JAVA: SAP_SLD_GUEST>,
<PRIV: ROLE: JAVA: NWA_JAVA_SUPPORT>
{ValidFrom=2030-01-01 12:00:00.0!!
ValidTo=2031-12-01 12:00:00.0}<PRIV: ROLE: JAVA: SAP_DI_DEVELOPER>
```

```
{M}<PRIV: JAVA: ONLY>|
<PRIV: SYSTEM: JAVA>|
{CTX=<BULGARIA>}<PRIV: ROLE: JAVA: SAP_SLD_GUEST>
|
{CTX=<GERMANY>}<PRIV: ROLE: JAVA: SAP_SLD_GUEST>|
<PRIV: ROLE: JAVA: SAP_SDI_C_SUPER_ADMINISTRATOR>|
{LINKID=12345!!
ValidTo=2030-06-06}<PRIV: ROLE: JAVA: NWA_JAVA_SUPPORT>
```

```
Deletes
{CTX=<CHINA>}<PRIV: ROLE: JAVA: SAP_SLD_GUEST>
Deletes
<PRIV: ROLE: JAVA: SAP_SLD_GUEST>
Modifies
<PRIV: ROLE: JAVA: NWA_JAVA_SUPPORT>, set
ValidTo to "2030-06-06 23:59:59.9"
Adds
<PRIV: ROLE: JAVA: SAP_SDI_C_SUPER_ADMINISTRATOR>
Future values are not modified:
{ValidFrom=2030-01-01 12:00:00.0!!
ValidTo=2031-12-01 12:00:00.0}<PRIV: ROLE: JAVA: SAP_DI_DEVELOPER>
```

```
<PRIV: JAVA: ONLY>,
<PRIV: SYSTEM: JAVA>,
{CTX=<BULGARIA>}<PRIV: ROLE: JAVA: SAP_SLD_GUEST>
,
{CTX=<GERMANY>}<PRIV: ROLE: JAVA: SAP_SLD_GUEST>,
{CTX=<CHINA>}<PRIV: ROLE: JAVA: SAP_SLD_GUEST>
```

```
{M}<PRIV: JAVA: ONLY>|
<PRIV: SYSTEM: JAVA>|
<PRIV: ROLE: JAVA: SAP_SLD_GUEST>|
{CTX=<FRANCE>}<PRIV: ROLE: JAVA: NWA_JAVA_SUPPORT>
```

```
Deletes
{CTX=<BULGARIA>}<PRIV: ROLE: JAVA: SAP_SLD_GUEST>
Deletes
{CTX=<GERMANY>}<PRIV: ROLE: JAVA: SAP_SLD_GUEST>
Deletes
{CTX=<CHINA>}<PRIV: ROLE: JAVA: SAP_SLD_GUEST>
Adds
<PRIV: ROLE: JAVA: SAP_SLD_GUEST>
Adds
{CTX=<FRANCE>}<PRIV: ROLE: JAVA: NWA_JAVA_SUPPORT>
```

{T} Operator

The {T} operator works the same way as the {M} operator, but in addition, it removes or keeps the attribute values with ValidFrom set in the future (if present).

Note

Attribute values provided with expired validity (ValidTo is set in the past) are ignored by the {T} operator and no action is performed on these values.

Reference attribute values provided with current validity (ValidFrom is set in the past and ValidTo is set in the future) are processed by the {T} operator as attribute values provided without any validity. That is, ValidTo is ignored.

Non-reference attribute values provided with current validity are processed by the {T} operator as attribute values with validity. Their ValidTo can be changed in the following way: {ValidTo=<new date or datetime>}ATTRIBUTE_VALUE.

For more information, see [3028156](#).

- To keep some future attribute value unchanged, you should provide the value together with its validity in the following way: {ValidFrom=<date1>!!ValidTo=<date2>}<Value>. If there is no ValidTo set, it could be skipped. For example: {ValidFrom=<date1>}<Value>
The date strings should be written following the ISO-8601 standard with the format: yyyy-MM-dd for date or yyyy-MM-ddTHH:mm:ss for datetime.

Note

If you use shorter date format, the Identity Management will automatically append a default time part which is: T00:00:00 for ValidFrom and T23:59:59 for ValidTo.

For example, if you have a privilege DEMO_PRIVILEGE assigned with ValidFrom=2030-12-01 00:00:00.0 and ValidTo=2035-12-01 23:59:59.9 to a user, and you want to keep this future assignment by using the {T} operator, you can use both valid syntaxes:
{ValidFrom=2030-12-01T00:00:00!!ValidTo=2035-12-01T23:59:59}DEMO_PRIVILEGE or
{ValidFrom=2030-12-01!!ValidTo=2035-12-01}DEMO_PRIVILEGE

Note

When reading the validity dates from the Identity Management database, in order to compare them with the validity strings provided in the To Identity Store pass, the {T} operator will ignore the nanoseconds from the time parts (if present). The compared time components are: hours, minutes, seconds.

- To keep a future context-based assignment unchanged using the {T} operator, you should provide the value together with its validity and, in addition, the context. See the following example: {CTX=<BULGARIA>!!ValidFrom=2030-12-01T15:15:31!!ValidTo=2035-12-01T23:00:05}DEMO_PRIVILEGE

Example

In this example, we modify the MXREF_MX_PRIVILEGE multivalue entry reference attribute in the To Identity Store pass.

Values in Identity Management database	Values in To Identity Store Pass	Result
<PRIV:JAVA:ONLY> with MSKEY = 30, <PRIV:SYSTEM:JAVA> with MSKEY = 31, {ValidFrom=2030-12-01 12:56:07.345!!ValidTo=2032-12-01 23:13:01.567}<PRIV:ROLE:JAVA:SAP_SLD_GUEST>	{T}30 31 <PRIV:ROLE:JAVA:NWA_JAVA_SUPPORT>	Deletes {ValidFrom=2030-12-01 12:56:07.345!!ValidTo=2032-12-01 23:13:01.567}<PRIV:ROLE:JAVA:SAP_SLD_GUEST> Adds <PRIV:ROLE:JAVA:NWA_JAVA_SUPPORT>

❖ Example

In this example, we modify the OFFICE_ADDITIONAL custom non-reference multivalue attribute in the To Identity Store pass.

Values in Identity Management data-base	Values in To Identity Store Pass	Result
ATTRIBUTE_VALUE_1 Pending Value Object 1 with attributes: MX_ATTRIBUTE_NAME = "OFFICE_ADDITIONAL", MX_ATTRIBUTE_VALUE = FUTURE_ATTRIBUTE_VALUE_ 1, MX_VALIDFROM = 2030-07-01T12:30:01, MX_VALIDTO = 2031-07-02T12:30:00	{T}ATTRIBUTE_VALUE_2 {ValidFrom=2030-07-01T1 2:30:01!! ValidTo=2031-07-02T12:3 0:00} FUTURE_ATTRIBUTE_VALUE_ 1	Deletes ATTRIBUTE_VALUE_1 Adds ATTRIBUTE_VALUE_2 Keeps the PVO holding FUTURE_ATTRIBUTE_VALUE_ 1

❖ Example

In this example, we modify the MXREF_MX_PRIVILEGE multivalue entry reference attribute in the To Identity Store pass. A context is provided for the entry reference attribute value.

Values in Identity Management data-base

Values in To Identity Store Pass

Result

```
<PRIV: JAVA: ONLY>,
<PRIV: SYSTEM: JAVA>,
<PRIV: ROLE: JAVA: SAP_SLD
_GUEST>,
{CTX=<BULGARIA>}<PRIV: R
OLE: JAVA: SAP_SLD_GUEST>
,
{CTX=<GERMANY>!!
ValidFrom=2030-12-01
12:56:07.345!!
ValidTo=2032-12-01
23:13:01.567}<PRIV: ROLE
: JAVA: SAP_SLD_GUEST>,
{CTX=<POLAND>!!
ValidFrom=2033-12-01
12:56:07.345!!
ValidTo=2034-12-01
23:13:01.567}<PRIV: ROLE
: JAVA: SAP_SLD_GUEST>,
{ValidFrom=2030-06-01
00:00:00.0!!
ValidTo=2031-06-01
23:59:59.9}<PRIV: ROLE: J
AVA: SAP_DI_DEVELOPER>,
{ValidFrom=2032-06-01
00:00:00.0!!
ValidTo=2033-06-01
23:59:59.9}<PRIV: ROLE: J
AVA: SAP_DI_DEVELOPER>,
<PRIV: ROLE: JAVA: NWA_JAV
A_SUPPORT>,
{ValidFrom=2030-06-01
00:00:00.0!!
ValidTo=2031-06-01
23:59:59.9}<PRIV: ROLE: J
AVA: NWA_JAVA_SUPPORT>,
{ValidFrom=2032-06-01}<
PRIV: ROLE: JAVA: NWA_JAVA
_SUPPORT>
```

```
{T}<PRIV: JAVA: ONLY>|
<PRIV: SYSTEM: JAVA>|
{CTX=<BULGARIA>}<PRIV: R
OLE: JAVA: SAP_SLD_GUEST>
|
{CTX=<GERMANY>!!
ValidFrom=2030-12-01T12
:56:07!!
ValidTo=2032-12-01T23:1
3:01}<PRIV: ROLE: JAVA: SA
P_SLD_GUEST>|
{ValidFrom=2033-12-01T1
2:56:07!!
ValidTo=2034-12-01T23:1
3:01}<PRIV: ROLE: JAVA: SA
P_SLD_GUEST>|
{ValidFrom=2030-06-01!!
ValidTo=2031-06-01}<PRI
V: ROLE: JAVA: SAP_DI_DEVE
LOPER>|
{ValidFrom=2032-06-01T0
0:00:00!!
ValidTo=2033-06-01T23:5
9:59}<PRIV: ROLE: JAVA: SA
P_DI_DEVELOPER>|
<PRIV: ROLE: JAVA: NWA_JAV
A_SUPPORT>|
{ValidFrom=2032-06-01}<
PRIV: ROLE: JAVA: NWA_JAVA
_SUPPORT>|
<PRIV: ROLE: JAVA: SAP_SDI
C_SUPER_ADMINISTRATOR>
```

```
Deletes
<PRIV: ROLE: JAVA: SAP_SLD
_GUEST>
Deletes {CTX=<POLAND>!!
ValidFrom=2033-12-01
12:56:07.345!!
ValidTo=2034-12-01
23:13:01.567}<PRIV: ROLE
: JAVA: SAP_SLD_GUEST>
Deletes
{ValidFrom=2030-06-01
00:00:00.0!!
ValidTo=2031-06-01
23:59:59.9}<PRIV: ROLE: J
AVA: NWA_JAVA_SUPPORT>
Adds
<PRIV: ROLE: JAVA: SAP_SDI
C_SUPER_ADMINISTRATOR>
```

Example

In this example, we modify the OFFICE_ADDITIONAL custom non-reference multivalue attribute in the To Identity Store pass.

Values in Identity Management data-base	Values in To Identity Store Pass	Result
ATTRIBUTE_VALUE_1, ATTRIBUTE_VALUE_2, ATTRIBUTE_VALUE_3, Pending Value Object 1 with attributes: MX_ATTRIBUTE_NAME = "OFFICE_ADDITIONAL", MX_ATTRIBUTE_VALUE = FUTURE_ATTRIBUTE_VALUE_ 1, MX_VALIDFROM = 2030-07-01T12:30:01, MX_VALIDTO = 2031-07-02T12:30:00 Pending Value Object 2 with attributes: MX_ATTRIBUTE_NAME = "OFFICE_ADDITIONAL", MX_ATTRIBUTE_VALUE = FUTURE_ATTRIBUTE_VALUE_ 1, MX_VALIDFROM = 2040-07-01, MX_VALIDTO = 2041-07-02 13:20:13.567 Pending Value Object 3 with attributes: MX_ATTRIBUTE_NAME = "OFFICE_ADDITIONAL", MX_ATTRIBUTE_VALUE = FUTURE_ATTRIBUTE_VALUE_ 2, MX_VALIDFROM = 2036-07-01 00:00:00.0	{T}ATTRIBUTE_VALUE_1 {ValidFrom=2036-07-01} FUTURE_ATTRIBUTE_VALUE_ 2 {ValidFrom=2040-07-01!! ValidTo=2041-07-02T13:2 0:13} FUTURE_ATTRIBUTE_VALUE_ 1 {ValidFrom=2050-01-01!! ValidTo=2052-01-01} FUTURE_ATTRIBUTE_VALUE_ 3	Deletes ATTRIBUTE_VALUE_2 Deletes ATTRIBUTE_VALUE_3 Deletes Pending Value Object 1 Adds {ValidFrom=2050-01-01!! ValidTo=2052-01-01} FUTURE_ATTRIBUTE_VALUE_ 3 // Will create new PVO for FUTURE_ATTRIBUTE_VALUE_ 3

Related Information

[Using Attribute Operators \(To Identity Store Pass\) \[page 337\]](#)

3.1.11.12.16.5 Modifying Properties of Assignments

When specifying the values of attributes in a To identity store pass, you can use a specific syntax to modify the properties of an assignment. The properties you can modify are validity and reason.

When modifying a property, you have to provide the LINKID as an operator. You retrieve the LINKID by reading it in the source of the pass with the LINKID operator.

If the assignment given by the LINKID has the status Failed or Rejected, the resulting operation will be a retry of the assignment, with the provided validity and/or reason.

If the status of the assignment is Pending or OK, the Modify validity task is executed, unless current executing task for this assignment is Validate add. In this case the validity is updated directly in the assignment, without triggering the Modify validity task. The reason is to be able to change validity during the approval process.

The generic syntax for specifying attribute properties is:

```
<AttributeName> {LINKID=<LinkID>!!<PropertyName>=<PropertyValue>!!...}<Value>
```

For instance:

```
MXREF_MX_ROLE {LINKID=3456!!VALIDFROM=2010-06-01!!VALIDTO=2010-12-31}<Manager>
```

<AttributeName>

Any attribute in the identity store.

<PropertyName>

This can either be one of the operators or properties.

Operators for Assignments

The operators are used to indicate a given behavior, and does not correspond to a given property.

Operator	Description
LINKID=xx	Set this operator to modify properties of the assignment with the given LinkID. If you are using an event task on an assignment (role or privilege) the pending value object will have the attribute MX_LINK_REFERENCE, which holds the LinkId. If you are configuring a job that retrieves assignments, you can use the LINKID operator on the "Source" tab to retrieve the LinkId of the assignments. If you for example specify: <pre>%{LINKID}MXREF_MX_ROLE%</pre> The returned data will look like this: <pre>{LINKID=123}456</pre>

Properties for Assignments

Property	Description
LinkReason	Reason for modifying the validity of this assignment.
ValidFrom/ValidTo	Validity period of the assignment.

<PropertyValue>

A legal value of the given property.

<Value>

The attribute value.

Examples

To modify the validity of a role and add a reason:

```
MXREF_MX_ROLE {LINKID=3456!!VALIDFROM=2010-06-01!!VALIDTO=2010-12-31!!  
LINKREASON=Extended deputyship}<Manager>
```

Related Information

[Using Attribute Operators \(To Identity Store Pass\) \[page 337\]](#)

3.1.11.12.17 Specifying the Destination for a To ASCII File Pass

When configuring a To ASCII file pass, you use the [Destination](#) tab to enter information about the ASCII file that is generated with this pass.

Context

To specify the destination, do the following:

Procedure

1. Select the [Passes](#) tab in the job view and select the pass.
2. Choose [Open](#) from the context menu to open the pass properties and select the [Destination](#) tab.
3. Enter the details about the destination file for this pass.

For details about the fields, see *Destination (To ASCII File Pass)*.

4. Use the [Definitions table](#) to enter definitions for the pass. The first line is the naming attribute.

Press `Ctrl` + `Space` to look up for the available options or to activate autocompletion.

ⓘ Note

You can use a template to simplify the work with the pass definitions ([Insert Template](#) button). For more information, see *Using a Template (To ASCII File Pass)*.

ⓘ Note

All variables and constants are enclosed in a % pair (%name%). If there is a % sign in the output, you must use the keyword \$PERCENT.

Option	Description
Prefix	Select one of the prefixes to control the behavior of the following definition. For more information about the available prefixes, see Prefix Options (To-Passes) [page 1074]
Attribute	Specify the column in the database. ⓘ Note Not all characters are allowed in attribute names. We recommend using the following characters: A-Z/a-z, 0-9 and _.
Value	Specifies the source for the value, or how the value is to be generated.

5. Choose **File** > **Save** or press `Ctrl` + `S` to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[Destination \(To ASCII File Pass\) \[page 1061\]](#)

[Using a Template \(To ASCII File Pass\) \[page 358\]](#)

3.1.11.12.17.1 Using a Template (To ASCII File Pass)

You can use a template to simplify work with definitions.

Context

Choose between the following methods to create the template:

- [Data source template](#)
- [Directory entry template](#)

Procedure

1. Select the [Passes](#) tab in the job view, then select and open a pass (choose [Open](#) from the context menu).
2. Select the [Destination](#) tab.
3. Choose [Insert Template](#) above the definitions table and select [Data source Template](#) to insert a template based on the data source.

The Identity Center tries to identify as much as possible of your data source to create a template you can use to make a CSV file (or other ASCII file). Each column in the data source corresponds to one line in the template.

4. Choose  [File](#)  or press `Ctrl` + `S` to save any changes made to the pass properties.

3.1.11.12.18 Specifying the Destination for a To Database Pass

When configuring a To Database pass, you use the [Destination](#) tab to enter information about the database you want to update.

Context

There are (at least) two ways you can create the database to which you are writing the data:

- Create the table "outside" the Identity Center. Make sure that the definitions match the columns in the table.
- Have the Identity Center create the table based on the definitions. If the table name specified in the [Database table name](#) field does not exist, the Identity Center will create the table using the specified definition. Make sure that the data types match the data types for your database. The default data types may not match all database types.

If the table exists, the Identity Center verifies whether the table definition matches the columns specified in the pass definition. New columns are added if needed using the specified name and data type. Columns are never removed and data types are never changed.

To specify the destination, do the following:

Procedure

1. Select the *Passes* tab in the job view and select the pass.
2. Choose *Open* from the context menu to open the pass properties and select the *Destination* tab.
3. Specify the connection to and other information about the destination database.

For details about the fields, see *Destination (To Database Pass)*.

4. Use the *Definitions table* to enter definitions for the pass. The first line is the naming attribute.

Press `Ctrl` + `Space` to look up for the available options or to activate autocompletion.

Note

You can use a template to simplify the work with the pass definitions (*Insert Template* button).

Note

All variables and constants are enclosed in a % pair (%name%). If there is a % sign in the output, you must use the keyword \$PERCENT.

Option	Description
<i>Prefix</i>	Select one of the prefixes to control the behavior of the following definition. For more information about the available prefixes, see Prefix Options (To-Passes) [page 1074]
<i>Attribute</i>	Specify the column in the database. Note Not all characters are allowed in attribute names. We recommend using the following characters: A-Z/a-z, 0-9 and _.
<i>Value</i>	Specifies the source for the value, or how the value is to be generated.

5. Choose **File > Save** or press `Ctrl` + `S` to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[Destination \(To Database Pass\) \[page 1063\]](#)

3.1.11.12.19 Specifying the Destination for a To Generic Pass

When configuring a To Generic pass, you use the [Destination](#) tab to enter information about the data repository you want to update.

Context

All data processing is handled by scripts, written in VBScript, JScript or PerlScript. The scripts are run in the following sequence:

1. The destination is opened by the script specified in the [Open destination](#) field.
2. For each entry in the temporary database, the script specified in the [Next data entry](#) field is run and the destination is updated using the contents of the [Definitions table](#).
3. If an [Auto-delete script](#) is defined on the [Delta](#) tab, this will be run for any deleted entries after all entries in the temporary database are processed.
4. The destination is closed using the script specified in the [Close destination](#) field.

To specify the destination of the pass:

Procedure

1. Select the [Passes](#) tab in the job view and select the pass.
2. Choose [Open](#) from the context menu to open the pass properties and select the [Destination](#) tab.
3. Specify the destination scripts. View and edit the scripts by choosing [Edit...](#) to the right of the fields. If no script is specified, choosing [Edit...](#) you can create a new script.

For details about the fields, see [Destination \(To Generic Pass\)](#).

4. Use the [Definitions table](#) to enter definitions for the pass. The first line is the naming attribute.

Press Ctrl + Space to look up for the available options or to activate autocompletion.

Note

You can use a template to simplify the work with the pass definitions ([Insert Template](#) button).

Note

All variables and constants are enclosed in a % pair (%name%). If there is a % sign in the output, you must use the keyword \$PERCENT.

Option	Description
Prefix	Select one of the prefixes to control the behavior of the following definition. For more information about the available prefixes, see Prefix Options (To-Passes) [page 1074]

Option	Description
<i>Attribute</i>	Specify the column in the database.
ⓘ Note Not all characters are allowed in attribute names. We recommend using the following characters: A-Z/a-z, 0-9 and _.	
<i>Value</i>	Specifies the source for the value, or how the value is to be generated.

5. Choose **File** > **Save** or press **Ctrl** + **S** to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[Destination \(To Generic Pass\) \[page 1064\]](#)

3.1.11.12.20 Specifying the Destination for a To LDAP Directory Pass

When configuring a To LDAP directory pass, you use the *Destination* tab to enter information about the directory you want to use.

Context

To specify the destination of the pass:

Procedure

1. Select the *Passes* tab in the job view and select the pass.
2. Choose *Open* from the context menu to open the pass properties and select the *Destination* tab.
3. Specify the details about the LDAP directory.

For details about the fields, see *Destination (To LDAP Directory Pass)*.

4. Use the *Definitions table* to enter definitions for the pass. The first line is the naming attribute.

Press **Ctrl** + **Space** to look up for the available options or to activate autocompletion.

Note

You can use a template to simplify the work with the pass definitions (*Insert Template* button).

Note

All variables and constants are enclosed in a % pair (%name%). If there is a % sign in the output, you must use the keyword \$PERCENT.

Option	Description
<i>Prefix</i>	Select one of the prefixes to control the behavior of the following definition. For more information about the available prefixes, see Prefix Options (To-Passes) [page 1074]
<i>Attribute</i>	Specify the column in the database. Note Not all characters are allowed in attribute names. We recommend using the following characters: A-Z/a-z, 0-9 and _.
<i>Value</i>	Specifies the source for the value, or how the value is to be generated.

5. Choose **File > Save** or press **Ctrl + S** to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[Destination \(To LDAP Directory Pass\) \[page 1065\]](#)

[Handling Multivalue Attributes \(To LDAP Directory/ To LDIF or DSML\) \[page 362\]](#)

3.1.11.12.20.1 Handling Multivalue Attributes (To LDAP Directory/ To LDIF or DSML)

When specifying the definitions for a To LDAP directory pass, you can specify which operation to perform on a specific attribute.

The attribute value is checked to see if it starts with one of the following character sequences:

- {A} = ADD attribute operation
- {R} = REPLACE attribute operation
- {D} = DELETE attribute operation

If the value does not start with one of these character sequences, it will be considered a REPLACE attribute operation.

When adding an LDAP entry, the attribute operations {A} and {R} will be ignored, and an attribute value starting with {D} will not be added.

3.1.11.12.21 Specifying the Destination for a To LDIF or DSML File Pass

When configuring a To LDIF or DSML file pass, you use the [Destination](#) tab to enter information about the LDIF file you want to update.

Context

To specify the destination of the pass:

Procedure

1. Select the [Passes](#) tab in the job view and select the pass.
2. Choose [Open](#) from the context menu to open the pass properties and select the [Destination](#) tab.
3. Specify the details about the LDIF file you want to update (or create a file in the DSML format).

For details about the fields, see *Destination (To LDIF or DSML File Pass)*.

4. Use the [Definitions table](#) to enter definitions for the pass. The first line is the naming attribute.

Press `Ctrl` + `Space` to look up for the available options or to activate autocompletion.

Note

You can use a template to simplify the work with the pass definitions ([Insert Template](#) button).

Note

All variables and constants are enclosed in a % pair (%name%). If there is a % sign in the output, you must use the keyword \$PERCENT.

Option	Description
Prefix	Select one of the prefixes to control the behavior of the following definition. For more information about the available prefixes, see Prefix Options (To-Passes) [page 1074]
Attribute	Specify the column in the database.

Option	Description
📌 Note Not all characters are allowed in attribute names. We recommend using the following characters: A-Z/a-z, 0-9 and _.	
<i>Value</i>	Specifies the source for the value, or how the value is to be generated.

5. Choose **File** > **Save** or press **Ctrl** + **S** to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[Destination \(To LDIF or DSML File Pass\) \[page 1067\]](#)

[Handling Multivalue Attributes \(To LDAP Directory/ To LDIF or DSML\) \[page 362\]](#)

3.1.11.12.22 Specifying the Destination for a To Custom Pass

When configuring a To Custom pass, you use the *Destination* tab to enter the necessary parameters for your Java class.

Context

To specify the destination of the pass:

Procedure

1. Select the *Passes* tab in the job view and select the pass.
2. Choose *Open* from the context menu to open the pass properties and select the *Destination* tab.
3. Specify the name of the Java class to be called in the *Pass type* field, and enter the parameter names and the corresponding values in the parameter table. These parameters are submitted to the class when it is called.

For details about the fields, see *Destination (To Custom Pass)*.

4. Use the *Definitions table* to enter definitions for the pass. The first line is the naming attribute.

Press **Ctrl** + **Space** to look up for the available options or to activate autocompletion.

ⓘ Note

You can use a template to simplify the work with the pass definitions (*Insert Template* button).

ⓘ Note

All variables and constants are enclosed in a % pair (%name%). If there is a % sign in the output, you must use the keyword \$PERCENT.

Option	Description
<i>Prefix</i>	Select one of the prefixes to control the behavior of the following definition. For more information about the available prefixes, see Prefix Options (To-Passes) [page 1074]
<i>Attribute</i>	Specify the column in the database. ⓘ Note Not all characters are allowed in attribute names. We recommend using the following characters: A-Z/a-z, 0-9 and _.
<i>Value</i>	Specifies the source for the value, or how the value is to be generated.

5. Choose ► *File* ► *Save* or press **Ctrl** + **S** to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[Destination \(To Custom Pass\) \[page 1068\]](#)

3.1.11.12.23 Specifying the Destination for a Shell Execute Pass

When configuring a Shell execute pass, you use the [Destination](#) tab to enter information about the command you want to run.

Context

To specify the destination of the pass:

Procedure

1. Select the [Passes](#) tab in the job view and select the pass.
2. Choose [Open](#) from the context menu to open the pass properties and select the [Destination](#) tab.
3. Specify the details about the command you want to run.

For details about the fields, see [Destination \(Shell Execute Pass\)](#).

Note

Be careful selecting both [Hide application/shell command](#) and [Wait for execution](#). If the application for any reason does not terminate as expected, you may need to close the application using Task Manager.

4. Use the [Definitions table](#) to enter definitions for the pass. The first line is the naming attribute.

Press `Ctrl` + `Space` to look up for the available options or to activate autocompletion.

Note

You can use a template to simplify the work with the pass definitions ([Insert Template](#) button).

Note

All variables and constants are enclosed in a % pair (%name%). If there is a % sign in the output, you must use the keyword \$PERCENT.

Option	Description
Prefix	Select one of the prefixes to control the behavior of the following definition. For more information about the available prefixes, see Prefix Options (To-Passes) [page 1074]
Attribute	Specify the column in the database.

Option	Description
	📌 Note Not all characters are allowed in attribute names. We recommend using the following characters: A-Z/a-z, 0-9 and _.
<i>Value</i>	Specifies the source for the value, or how the value is to be generated.

5. Choose **File** > **Save** or press **Ctrl** + **S** to save any changes made to the pass properties. Select another tab to change other parameters, if not yet defined.

Related Information

[Destination \(Shell Execute Pass\) \[page 1069\]](#)

3.1.11.12.24 Calling Scripts in To-Passes

Scripts take one input parameter, and return a text. If it is necessary to use more input parameters, this can be done by combining strings.

Context

Combining of strings can be done in the following way:

Procedure

You may for instance use **&** as a delimiter, and then split the strings again inside the script. E.g.

```
Func (%dn%&%cn%)
```

Syntax: \$function.<name>(<parameter>)\$\$

3.1.11.12.25 Using Indirect DN Lookups in To-Passes

When specifying the definitions for a To-pass, you may need to combine information from different data sources into the same entry in the directory. This can be done in different ways, depending on the nature of the data.

Context

If you can identify the DN directly, you can add attributes to the entry. If not, you have to find the correct DN before you can add the attribute. This can be done using an LDAP URL. For example, if you have the telephone number (in the temporary database column 'phone'), you can use the following syntax:

Attribute	Value
dn	\$ldap://host/dn?sub??telephone=%phone%

In most cases, searching all entries sequentially for the correct DN would have exceeded the time and size limits. Using indirect DN lookup to locate the correct entry solves this problem.

Note

The attribute you use as the key must be unique. If there is more than one match, only the first one is returned. You are not able to control this.

3.1.11.12.26 Inserting Variables and Constants

Variables and constants can be used in several places in the user interface and from scripts.

- Using variables and constants in the User Interface
When you enter the pass definition, the easiest way to insert variables or constants is to choose them from the list of available variables and constants. To do this, select a row and press **Ctrl** + **Space**. This will show a complete list of all available variables that can be inserted. Code completion is enabled.
- Using variables and constants from scripts
Most constants can be referenced from scripts using the syntax described below. Additionally, you can use the function `uGetConstant` to retrieve the value of a given constant.

Variables and Constants	Description	Syntax
Recordset variables	Recordset variables are used to reference data source variables in the <i>Definitions</i> field of To-passes. The runtime engine will replace the variable with the value when running the job.	Code Syntax <pre>%variable%</pre> For example: %givenname% %surname%
Job constants	Job constants have the same syntax as the recordset variables, but an additional \$ sign must be included. Job constants can be used in any field in the pass properties.	For example: cn=%name%, %\$STOPDN%
Package constants	Package constants are only available within the package. You can call them from jobs, tasks and scripts within the package.	Code Syntax <pre>;%pck.<constant>%</pre>
Repository constants	The values for the repository constants are used to access the data source. Note - If you have a standard job and you have configured a repository for a job pass, use the uGetRepositoryName function to access the name of that repository. - If you have a repository job, you can access the name of the job repository by using ;%\$rep.\$NAME%. - If you have a repository job with a pass that has a different repository, you can access the repository name of the job either way (uGetRepositoryName or ;%\$rep.\$NAME%).	Code Syntax <pre>;%\$rep.<constant>%</pre>
System parameters	System parameters behave exactly like the job constants, except that they are predefined by the Identity Center. They can be used in any field in the pass properties.	For example: ;%\$ddm.actionid%

3.1.11.12.27 Using System Parameters

System parameters behave exactly like the job constants, except that they are predefined by the Identity Center. The table below lists the system parameters available.

The system parameters work in all fields.

Parameter Name	Description
\$ddm.actionid	TaskID of the current task. Only relevant for the Identity Center, and when the job is running as an action task.
\$ddm.databasetype	The database system of the current Identity Management database. (Only relevant for the Identity Center.) 1 = Microsoft SQL Server 2 = Oracle
\$ddm.date	Current date in the format specified in the Regional settings.
\$ddm.date8601	Returns the current date in the following format: yyyy-mm-dd
\$ddm.datetime8601	Returns the current date and time in the following format: yyyy-mm-ddThh:mm:ss
\$ddm.-date	Current date with hyphens (-) as delimiters. Useful when generating file names, for instance.
\$ddm.errcount	The number of errors that occurred.
\$ddm.job	Each time a job runs, the job number for that specific job is incremented. This job number can be very useful, making it possible to use SQL statements on the Delta databases created.
\$ddm.lastdate	The date and time when the job was last run in the following format: dd/mm/yyyy hh:mm .
\$ddm.mcjob	JobID of the current job. (Only relevant for the Identity Center.)
\$ddm.identitycenter	The connection string to the Identity Management database. (Only relevant for the Identity Center.)
\$ddm.path	Path to the location of the runtime executable (DSERT.EXE or DSE.jar).
\$ddm.repository	Returns the current repository id.

Parameter Name	Description
\$ddm.time	Returns the current time in the following format: hh:mm:ss.
\$ddm.time24	Returns the current time in the following format: hh:mm:ss
\$env.<ENVNAME>	Returns the value of a system variable or property specified in <ENVNAME>. For example: \$env.os.version - Returns the version of the operating system. For example: 6.1 \$env.user.dir - Returns the user working directory. It is the directory where java was run from. For example: C:\usr\sap\idm\Identity Center \$env.file.separator - Returns the character that delimits file and directory names. For example: \ \$env.user.home - Returns the profile folder used by the Windows System account: C:\Windows\System32\config\systemprofile \$env.java.home - Returns the JRE installation path. For example: D:\sapjvm\sapjvm_8\jre \$env.user.name - Returns the name of the current user.
\$ddm.PERCENT	Use this system parameter to insert a percent character that should be written to the target.
\$ddm.PIPE	Use this system parameter to insert a pipe character () that should be written to the target.

3.1.11.12.28 Including Properties when Reading Attribute Values from the Identity Store

In a To-pass that has the identity store as a source, you can use a specific syntax in the definition of the pass to retrieve properties from attributes.

There are different properties for reference attributes (role/privilege assignments, manager reference, owner) and regular attributes (last name, title etc). Examples of properties are validity period or a context. The returned values are on the same format as when writing properties of attribute values to the identity store. This means that this also can be used in a To identity store pass with the identity store as source.

The generic syntax for specifying attribute properties is:

Code Syntax

```
%{<PropertyName>!!<PropertyName>!!...}<AttributeName>%
```

For example:

Description	Example syntax	Return value
Retrieving roles including the corresponding expiry time.	↔ Code Syntax % {VALIDTO}MXREF_MX_ROLE%	↔ Code Syntax {VALIDTO=2010-06-30}123
Retrieving all roles, including validity and the MSKEY-VALUE for all roles.	↔ Code Syntax % {VALIDTO!! VALIDFROM!! MSKEYVALUE}MXREF_MX_ROLE%	↔ Code Syntax {VALIDTO=2010-12-31!! VALIDFROM=2007-12-01!! MSKEYVALUE=RoleManager}123 {VALIDTO=!!VALIDFROM=2000-03-01!! MSKEYVALUE=RoleAccounting}234
Retrieving all role assignments including contexts.	↔ Code Syntax % {CTX}MXREF_MX_ROLE%	↔ Code Syntax {CTX=42}123 {CTX=43}123 {CTX=44}123

<PropertyName>

This can either be one of the operators or properties. The operators are used to indicate a given behavior, and does not correspond to a given property.

Reference attributes

Operators

Operator	Description
LINKID	If this is set, the returned values are LinkIds instead of MSKEYs. This cannot be combined with any other operators.
LINKSTATE=x	Used as a filter to indicate which references (links) to return. By default all active references are returned, but you can specify which references to return: -1 : All references 0 : Active references (default) 1 : Inactive references 2 : Deleted references

Properties

Property	Description
AuditId	Returns the reference to the audit ID which created this relation. Not set for inherited assignments, only direct assignments to this user. All assignments made in the same request, will have the same AuditID.
CTX	Reference to a context for this role/privilege assignment. Note This property can only be written when creating a reference attribute, not when modifying it.
LinkAdded	Returns the date/time when the relation was added.
LinkAssigner	Returns the MSKEY of the user adding the assignment.
LinkExecState	Returns the state of this assignment.
LinkExecStateHier	Returns the accumulated state of this hierarchy assignment.
LinkGroupGuid	Returns the GUID used for grouping of assignments. All assignments handled within the same group will have the same GUID.
LinkModified	Returns the date/time when the link was last modified.
LinkModValidFrom	Returns the modified value for ValidFrom. This can be retrieved from the Validate modify validity task to determine if the new value can be accepted.
LinkModValidTo	Returns the modified value for ValidTo. This can be retrieved from the Validate modify validity task to determine if the new value can be accepted.
LinkReason	Returns the reason specified for this assignment.
LinkState	Returns the state of the relation: 0: Active. Indicates that the assignment is active, and should be visible to any application viewing the assignments. 1: Inactive. An inactive relation may for example be a privilege which is assigned, but the add member process has not yet completed. 2: Deleted link. The link has been deleted.

Property	Description
LinkType	Returns the type of relation: 0: Unspecified entry reference (i.e. manager, secretary). 1: Hierarchy (i.e. role -> role or privilege -> role). 2: Member (i.e. user -> role, user -> privilege, user -> group). 3: Owner (i.e. the MX_OWNER attribute).
MSKEYVALUE	If this is set, the MSKEYVALUE of the entry referenced with this relation will be returned as a property.
ProcessInfo	Returns the ProcessInfo specified for the attribute. The syntax is the same as for UserId.
Repository	Returns a reference to the repository that is defined as owner.
RequestId	Returns the assignment's RequestID. This is a general purpose request identifier, used for instance by the integration with SAP BusinessObjects Access Control. The RequestID is used by the Request complete task defined for the identity store.
ValidFrom/ValidTo	Returns the validity period of the assignment.

Regular attributes

The operators are used to indicate a given behavior, and does not correspond to a given property.

Operators

Operator	Description
FutureValues	If this is set, all future pending values (i.e. stored in the MX_PENDING_VALUE entries) for this attribute will be retrieved.

Properties for regular attributes

Property	Description
AuditId	Returns the Audit ID of the task which last modified this attribute.
ChangedBy	Returns the MSKEY of the user who last changed this attribute.

Property	Description
ChangeNumber	Returns the change number for this attribute. The change number is incremented every time a change is done, and is global to each Identity Center database.
ChangeType	Returns the operation which changed this attribute. This can of course be combined with the occurrence attribute, to retrieve old operations. This can have one of the following values: 0: Modify 1: Delete 2: Add 3: Disabled 4: Enabled
Checksum	Returns the checksum of the attribute value.
ModifyByMsKeyValue	Returns the MSKEYVALUE of the user which modified the entry. Done by looking up the ChangedBy column.
ModifyTime	The date/time the attribute was modified.
NumHistory	Number of historic values for this attribute.
Repository	Returns a reference to the repository that is defined as owner.
UserId	Returns the UserId string of the attribute.
ValidFrom/ValidTo	Validity period of the assignment.

<AttributeName>

Any attribute in the identity store. If the entry is a pending value object, you can use a specific syntax to retrieve attributes from the entries referenced from the pending value object.

3.1.11.12.29 LinkExecState and LinkExecStateHier

When retrieving properties of reference attributes in the identity store, you can include the status of the member event processes processing the assignments/deassignments.

LinkExecState

This is the status of a given assignment. The columns in the table contain the following information:

- **Return value** - The numeric value that is returned as the property of the attribute. The first value is that status returned from an Add member process while the other value is the status returned from a Remove member process. In this case *Assignment OK* means that the Remove member process completed successfully (*Deassignment OK*).
- **Text** - This is the status as it is displayed in the Identity Management User Interface.
- **Description** - Textual description of the status.

See the possible values in the table below:

Return value (add/remove)	Text (displayed in User Interface)	Description
0/1024	OK	This is the default value, and will be set in the case where there are no member event processes defined.
1/1025	OK	Assignment OK.
2/1026	Rejected	Assignment rejected.
4/1028	Failed	Assignment failed.
512/1536	Pending add/Pending remove	Assignment still in process.

LinkExecStateHier

This is the accumulated status for a given assignment and all inherited assignments that are executed as a result of this assignment. The columns are the same as in the previous table.

See the possible values in the table below:

Return value (add/remove)	Text (displayed in User Interface)	Description
0/1024	OK	This is the default value, and will be set in the case where there are no member event tasks defined.

Return value (add/remove)	Text (displayed in User Interface)	Description
1/1025	OK	All assignments OK.
2/1026	Rejected	All assignments rejected.
3/1027	Partial OK	Some assignments rejected. Some assignments OK.
4/1028	Failed	All assignments failed.
5/1029	Failed	Some assignments failed. Some assignments OK.
6/1030	Failed	Some assignments rejected. Some assignments failed.
7/1031	Failed	Some assignments rejected. Some assignments failed. Some assignments OK.
512/1536	Pending add/Pending remove	Assignments still in process.
513/1537	Pending add/Pending remove	Some assignments OK. Assignments still in process.
514/1538	Pending add/Pending remove	Some assignments rejected. Assignments still in process.
515/1539	Pending add/Pending remove	Some assignments rejected. Some assignments OK. Assignments still in process.
516/1540	Pending add/Pending remove	Some assignments failed. Assignments still in process.
517/1541	Pending add/Pending remove	Some assignments failed. Some assignments OK. Assignments still in process.
518/1542	Pending add/Pending remove	Some assignments rejected. Some assignments failed. Assignments still in process.

Return value (add/remove)	Text (displayed in User Interface)	Description
519/1543	Pending add/Pending remove	Some assignments rejected. Some assignments failed. Some assignments OK. Assignments still in process.

3.1.11.12.30 Including Attributes from Referenced Entries when Processing Pending Value Objects

An attribute in the identity store is always of a given entry type. If the attribute's entry type is pending value object (MX_PENDING_VALUE), you can reference attributes from the involved entries.

If the pending value object contains future values for a regular attribute, for instance an address or a title that should be valid in the future, you can retrieve other information about the user using the following syntax:

Code Syntax

```
PARENT.<AttributeName>
```

Note

If *Retrieve attributes from pending value* check box on the *Source* tab of the pass is not selected, you will only be able to access attributes from the parent entry. If the check box is selected, referencing attributes without a prefix reads from the pending value, while using the PARENT. prefix reads from the parent entry.

If the pending value object is created as a result of a member event process, you can retrieve information about the entries involved in the assignment:

Code Syntax

```
SUBJECT.<AttributeName>
TARGET.<AttributeName>
CONTEXT.<AttributeName>
LINK.<Property>
```

Where SUBJECT is normally the user and TARGET the role or privilege being assigned. CONTEXT is the context reference, and LINK is the relation itself. <Property> refers to the properties for reference attributes. You can reference these properties in both ways.

Example

```
PARENT.MX_TITLE
```


If the pending value object contains future values for the title of a person, this will return the current title.

SUBJECT.DISPLAYNAME

TARGET.DISPLAYNAME

If the pending value object is created as a result of a role assignment, this will return the display name of the user and the role.

3.1.11.12.31 Configuring Delta for Passes

In the *Delta* tab of the pass properties, you may enable and configure delta for the passes.

Context

To configure delta for a pass, do the following:

Procedure

1. Select the *Passes* tab in the job view.
2. Select the pass and choose *Open* from the context menu.
3. Select the *Delta* tab.
4. Enable the delta for the pass. Define the delta database, delta identifier and other delta properties.

Note

Automatic deletion properties are relevant for To-passes and not for the From-passes. The exception is the following To-passes:

- To ASCII file pass
- To Database pass when the *SQL updating* option is enabled
- Shell execute pass

Note

For the To Generic pass, the automatic deletion option *Autodelete script* is relevant.

5. Choose **File** > **Save** or press **Ctrl** + **S** to save the changes. Select another tab to change other parameters, if not yet defined.

Related Information

[Delta \(From-Passes\) \[page 1047\]](#)

[Delta \(To-Passes\) \[page 1070\]](#)

[The Delta Database \[page 1078\]](#)

3.1.11.12.32 Entering a Description of a Pass

You can document a pass in the [Documentation](#) tab of the pass properties.

Context

To enter a description of a pass, do the following:

Procedure

1. Select the [Passes](#) tab in the job view and select the pass.
2. Choose [Open](#) from the context menu to open the pass properties and select the [Documentation](#) tab.
3. Select the [Documentation](#) tab.
4. Enter the description of the pass.
5. Choose [File](#) > [Save](#) or press `Ctrl` + `S` to save the entered job description.

Related Information

[Documentation \(Passes\) \[page 1031\]](#)

3.1.11.13 Maintaining Scripts

You can add scripts to your jobs. Scripts can be used for the following purposes:

- To correct inconsistent source data, such as different telephone number formats
- To test source data and act on the results
- To format data
- To convert between character sets

- To split composite data elements into separate parts
- To combine several data elements into one parameter

When using the Java runtime engine, you have to write your scripts in JavaScript. JavaScript is a common name for a scripting language defined in ECMA standard 262.

📌 Note

All scripts in one job must be written using the same script language. You cannot mix languages in the same job.

→ Remember

The syntax of the scripts is validated by the external library JSHint, and for that reason the displayed warnings cannot be influenced by SAP Identity Management.

Parameters

All scripts receive one string parameter and are supposed to return a string parameter.

📌 Note

This does not apply to the scripts used by the From Generic and To Generic passes. The scripts used by From Generic passes return a dictionary object, and the scripts used in To Generic passes receive a dictionary object.

If it is necessary to use more input parameters, you can do this by combining strings. You can use & as a delimiter for example, and then split the string inside the script.

For instance:

```
Func(%dn%&%cn%)
```

If you are calling a function (uFunction or user defined function) and have a parameter containing a quote ("), this must be escaped, using the backslash.

For example:

```
MyFunction ("The string \"abc\" contains no d")
```

Pre-defined script names

There are two pre-defined script names:

- INITIALIZE
- TERMINATE

Note

When using the Java runtime engine, the script names are case sensitive.

A script named INITIALIZE is always run at the start of a job. A script named TERMINATE is always run at the end of a job. These scripts can be used to pre-process or post-process data, produce a job report or similar tasks. You can also preload program objects (for example, Microsoft Word files) for later use to avoid overhead every time a script is called.

Related Information

[Standard ECMA-262](#) 

3.1.11.13.1 Adding a Script to a Job

You can add scripts to your jobs.

Prerequisites

- The package is checked out.
- The job is opened in the job perspective.

Context

To add a script to a job:

Procedure

1. Select the *Scripts* tab in the job view.
2. Choose *New* from the context menu.
The *New script* dialog box is displayed.
3. Enter a name for the script and choose *OK*.

A template for the script is displayed in the script editor view.

3.1.11.13.2 Linking to a Package Script

You can create a link to a package script.

Prerequisites

- The package is checked out.
- The job is opened in the job perspective.

Context

To create a link to a package script, proceed as follows:

Procedure

1. In the job view, select the *Scripts* tab.
2. Choose *Add Link to Package Script* from the context menu.
3. In the *Connection to Package Script* dialog box, you can do the following:
 - In the *Select Package* field, select a package.
 - In the *Filter* field, filter the name of the package script.
 - In the *Select Script* field, select the package script to link to.
4. Choose *OK*.

3.1.11.13.3 Modifying a Script

Prerequisites

- The package is checked out.
- The job is opened in the job perspective.

Context

To modify a script, proceed as follows:

Procedure

1. In the job view, select the [Scripts](#) tab and open the script that you want to modify.
The script code is displayed in the script editor view.
2. Modify the content of the selected script.
3. Press `Ctrl` + `Space` to look up for available functions and constants.

3.1.11.14 Adding or Modifying a Job Constant

Job constants make it easy to change parameters for all passes in one operation.

Context

Job constants can be used in any field in the pass definitions. They are stored in the job definition file.

Examples of job constants:

- The DN (distinguished name) of the top level of the data structure in the directory
- Paths in the file system

Job constants can be referenced from scripts using the syntax described below. Additionally, you can use the function [uGetConstant](#) to retrieve the value of a given constant. Job constants can be used in any field in the pass properties.

Job constants have the following syntax: `cn=%name%, %$TOPDN%`

To add or modify a job constant, proceed as follows:

Procedure

1. Select the [Constants](#) tab in the job view.
2. Choose [New](#) from the context menu.
The [Job Constant](#) dialog box is displayed.
3. Enter values in the fields. For more information about the fields, see [Job Constant](#).
4. Choose [OK](#) to save the job constant and close the dialog box.

Related Information

[Constants \(Job\) \[page 1083\]](#)

3.1.11.15 Adding or Modifying a Job Variable

Job variables contain values that are common to all passes in a job and are available only from scripts using the internal functions `uGetUserVar` and `uSetUserVar`. They are stored in a `<job name>.ddx` file in the same directory as the job file.

Context

To add or modify a job variable, proceed as follows:

Procedure

1. Select the [Variables](#) tab in the job view.
2. Choose [New](#) from the context menu.
The [Job Variable](#) dialog box is displayed.
3. Enter variables in the fields. For more information about these fields, see [Job Variable](#).
4. Choose [OK](#) to close the dialog box.

Related Information

[Adding or Modifying a Job Constant \[page 384\]](#)

[uGetUserVar \[page 1172\]](#)

[uSetUserVar \[page 1223\]](#)

[Variables \(Job\) \[page 1085\]](#)

3.1.11.16 Maintaining Dispatchers

Dispatchers are responsible for executing tasks and jobs in the system.

Maintaining the dispatchers in the Identity Management Developer Studio includes operations like viewing and configuring, or modifying, the dispatcher properties, viewing the dispatcher status and stopping the dispatchers.

Related Information

[Dispatchers \[page 67\]](#)

3.1.11.16.1 Getting a Dispatcher Overview and Viewing their Status

The dispatcher overview lists all dispatchers in this Identity Center and displays information about them.

Context

To view the list of dispatchers and see their statuses, do the following:

Procedure

1. Select the [Dispatchers](#) folder in the tree view (under [Management](#) in your Identity Center).
2. Choose [Open](#) from the context menu.

Dispatcher overview is displayed. The list of dispatchers contains the following information (columns):

Field	Description
Dispatcher (name)	Displays the name of the dispatcher.
Version	Displays the version number of the dispatcher. It is automatically updated by the dispatcher when connecting to the Identity Center.
Status	Displays the dispatcher's service state, e.g. Running or Stopped .
Last check	Time for the dispatcher's last check (when it was last checked for scheduled jobs in the Identity Center).
Tasks	Indicates whether the dispatcher processes tasks.
Approvals	Indicates whether the dispatcher processes approvals.
Relations	Indicates whether the dispatcher evaluates relations.
Housekeeping	Indicates whether the dispatcher performs housekeeping actions.
Run jobs	Indicates whether the dispatcher runs jobs.

3. Use the [Search](#) field to search for selected dispatchers in the list.

4. To continuously update the log, select [Automatically refresh every <X> seconds](#) and enter the number of seconds' interval between each refresh (by default 30 seconds). To manually update the dispatcher overview list, choose the [Refresh](#) button.

3.1.11.16.2 Viewing and Configuring Dispatcher Properties

You can configure properties for the dispatchers.

Context

To view and configure (or modify) a dispatcher and its properties, do the following:

Procedure

1. Select and expand the [Dispatchers](#) folder in the tree view (under [Management](#) in your Identity Center).
2. Select the dispatcher you want to configure, then choose [Open](#) from the context menu.
3. Select a tab to enter or view the information about the dispatcher.
 - [General](#)
 - [Policy](#)
 - [Jobs](#)
 - [Advanced](#)
4. Choose [File](#) > [Save](#) or press `Ctrl` + `S` to save the changes.

Related Information

[Dispatcher Properties \[page 974\]](#)

[General \(Dispatcher\) \[page 975\]](#)

[Advanced \(Dispatcher\) \[page 979\]](#)

[Policy \(Dispatcher\) \[page 977\]](#)

[Jobs \(Dispatcher\) \[page 978\]](#)

3.1.11.16.3 Specifying General Properties of a Dispatcher

When configuring a dispatcher, you use the [General](#) tab to specify information about the dispatcher.

Context

Perform the following steps:

Procedure

1. Select and expand the [Dispatchers](#) folder in the tree view (under [Management](#) under your Identity Management database node).
2. Select the dispatcher that you want to configure and choose [Open](#) from the context menu.
3. Select the [General](#) tab in order to enter the information about the dispatcher. You can perform the following operations here:
 - Change the dispatcher name.
 - Add information about the runtime engines for the dispatcher.
 - Define the log level for the dispatcher.
 - View the version of the dispatcher, when it was last checked for scheduled jobs, and see the service status.
 - Suspend a running dispatcher by pressing [Suspend](#).
 - Resume a suspended dispatcher by pressing [Resume](#).
 - Stop the dispatcher by pressing [Stop Dispatcher](#).
4. Press [Refresh](#) in order to update the information.
5. Choose ► [File](#) ► [Save](#) or press Ctrl + S to save the changes. Press [Advanced](#) to open the [Advanced](#) tab and configure the custom parameters for the dispatcher.

Related Information

[General \(Dispatcher\) \[page 975\]](#)

3.1.11.16.4 Configuring Custom Parameters for a Dispatcher

When configuring a dispatcher, you use the [Advanced](#) tab to configure custom parameters for the dispatcher. The [Advanced](#) tab is only accessible from the [General](#) tab of the dispatcher properties.

Context

ⓘ Note

The dispatcher needs to be started before the custom parameters and their default values are filled in.

ⓘ Note

Do not change these parameters unless you are sure about the consequences.

To configure custom parameters:

Procedure

1. Select and expand the [Dispatchers](#) folder in the tree view (under [Management](#) in your Identity Center).
2. Select the dispatcher you want to configure, then choose [Open](#) from the context menu.
3. Select the [General](#) tab, and then choose the [Advanced](#) button to open the [Advanced](#) tab and configure the custom parameters for the dispatcher.
4. Select the [Advanced](#) tab.
5. Change the parameters, and then choose [File](#) > [Save](#) or press `Ctrl` + `S` to save the changes.

Related Information

[General \(Dispatcher\) \[page 975\]](#)

[Advanced \(Dispatcher\) \[page 979\]](#)

3.1.11.16.5 Configuring the Dispatcher Policy

When configuring a dispatcher you use the [Policy](#) tab to configure the behavior of the dispatcher to optimize performance on the system.

Context

Which configuration gives the most optimal performance will vary depending on a number of factors and must be based on experience with the specific system.

Note

If you change some of these settings, make sure that at least one dispatcher performs each of them.

To configure the dispatcher task execution policy, do the following:

Procedure

1. Select and expand the [Dispatchers](#) folder in the tree view (under [Management](#) in your Identity Center).
2. Select the dispatcher you want to configure, then choose [Open](#) from the context menu.
3. Select the [Policy](#) tab and configure the task execution policy for the dispatcher. You may:
 - define the interval between each time the dispatcher should check for scheduled jobs.
 - define if the dispatcher should process tasks, recalculate relations (assignments), process approvals and/or execute scheduled procedures for housekeeping.
 - define the housekeeping interval.
 - specify that the dispatcher should start the runtime engine to run jobs, and if the dispatcher should check for the provisioning and/or regular jobs for the specified runtime engine.
 - define log level, stack trace and Java options for the Java runtime engine.
4. Choose  [File](#)  [Save](#) or press `Ctrl` + `S` to save the changes.

Related Information

[Policy \(Dispatcher\) \[page 977\]](#)

3.1.11.16.6 Specifying the Jobs of a Dispatcher

When configuring a dispatcher you use the [Jobs](#) tab to specify which jobs this dispatcher is responsible for executing.

Context

To specify the jobs for the dispatcher, do the following:

Procedure

1. Select and expand the [Dispatchers](#) folder in the tree view (under [Management](#) in your Identity Center).
2. Select the dispatcher you want to configure, then choose [Open](#) from the context menu.
3. Select the [Jobs](#) tab and define the jobs to be run by the dispatcher.

All jobs in the Identity Center are listed.

4. Select the jobs that can be run by the dispatcher.
5. Choose  [File](#)  or press `Ctrl` + `S` to save the changes.

Related Information

[Jobs \(Dispatcher\) \[page 978\]](#)

3.1.11.16.7 Suspending and Resuming a Dispatcher

You can suspend a dispatcher for a period of time and resume it again when needed.

Context

To suspend or resume a dispatcher, proceed as follows:

Procedure

1. Select and expand the *Dispatchers* folder in the tree view (under *Management* under your Identity Management database node).
2. Select the dispatcher that you want to suspend or resume and choose *Properties* from the context menu.
3. Select the *General* tab.
 - To suspend the dispatcher, press *Suspend*.
 - To resume the dispatcher, press *Resume*.

Note

If the process of suspending or resuming the dispatcher takes a long time (a minute or more), something might have gone wrong. If this is the case, you will be notified and prompted to restart your dispatcher. To start a dispatcher that has been stopped, use the Identity Management Dispatcher Utility.

Related Information

[General \(Dispatcher\) \[page 975\]](#)

3.1.11.16.8 Stopping a Dispatcher

Dispatchers are stopped by a signalling flag that is set in the database. This flag is set from the [General](#) tab in the dispatcher properties.

Context

To stop the dispatcher, do the following:

Procedure

1. Select and expand the [Dispatchers](#) folder in the tree view (under [Management](#) under your Identity Management database node).
2. Select the dispatcher that you want to stop and choose [Properties](#) from the context menu.
3. Select the [General](#) tab.
4. Stop the dispatcher by pressing [Stop Dispatcher](#).

Note

To start the dispatcher again, use the Identity Management Dispatcher Utility.

Related Information

[General \(Dispatcher\) \[page 975\]](#)

3.1.11.17 Copying Jobs

You can copy and paste jobs from one package to another within one identity store or multiple identity stores.

Context

When you copy a job, you copy all the passes, scripts, package script references, and constants defined for this job. You cannot copy the job variables defined for this job.

To copy and paste a job, proceed as follows:

Procedure

1. In the Identity Management Developer Studio tree view, select your identity store.
2. Select the package that contains the job you want to copy.
3. Select the job and choose *Copy* from the context menu.
4. Select the package where you want to paste the job.

Note

You need to check out the package where you want to paste the job.

5. Select the *Jobs* folder and choose *Paste* from the context menu.

Results

You have copied a job with all passes, scripts, package script references, and constants defined for this job. All public package scripts are automatically relinked. All private package scripts are displayed with a warning sign. You can relink them manually by choosing an option from the listed below:

- Make the private package script public from the source package.
- Create a job script in the target job and copy the content of the private package script in it.

3.1.12 Searching for an Object in the Configuration

You can search for objects like jobs and tasks in the configuration and locate the object from the search result.

Context

To search for an object:

Procedure

1. **Note**

The highest level you can start your search from is the identity store node.

Select the node in the tree view where you want to start the search and choose [Search...](#) from the context menu.

The [Search](#) dialog box appears.

2. Enter the text to search for. Use ***** or **?** as wildcards. The search is performed on the ID and name of the objects.
3. Select which type of object you want to search for:
 - [Process/Task/Form](#)
 - [Job](#)
 - [Package](#)
 - [Package Constant](#)
 - [Package Script](#)

4. Choose [OK](#).

The search result is displayed in the [Search](#) view.

Note

Only objects within packages you have access to, are listed.

5. If the search result contains a long list, you can limit the list to the entries corresponding to the value you specify in the [Filter](#) field.
6. You can open the object, by selecting it and choosing [Open](#) from the context menu.
7. You can locate the object in the configuration, by selecting it and choosing [Locate](#) from the context menu. The configuration tree is expanded to show the selected entry. If it is a task within a process, the task flow editor is opened and the task is highlighted.
8. Choose [Refresh](#) from the context menu to perform the search again and update the search result.

3.1.13 Locating Parent Objects in the Tree View

When working with jobs, processes and forms in multiple views, you can locate the node in the tree view corresponding to the selected view.

Context

To locate the view's parent node in the tree view:

Procedure

1. Select a view for one of the objects.
2. Select [Locate](#) from the properties view menu.

In the tree view, you can locate the node corresponding to the selected view.

3.1.14 Closing Related Views for an Object

When working with jobs and processes, you may end up with several views for each object. You can close all views for an object or close all views for all other objects.

Context

To locate a related object or close open views:

Procedure

1. Select a view for one of the objects.
2. Locate the object by selecting [Locate](#) from the properties view menu.
The corresponding node in the tree view is selected, and if the selected view belongs to a job or a pass, the job is brought to front in the job perspective.
3. Close all related views by selecting [Close Related Objects](#) from the properties view menu.
You can also select a node in the tree view and choose [Close Related Objects](#) from the context menu or a job in the job perspective and choose [Close Related Objects](#) from the properties view menu.
4. Close all views not related to the selected object, by selecting [Close Not Related Objects](#) from the properties view menu.
You can also select a node in the tree view and choose [Close Not Related Objects](#) from the context menu or a job in the job perspective and choose [Close Not Related Objects](#) from the properties view menu.

3.1.15 Changing the Display Language of Attributes, Entry Types and Forms

Context

Note

As of SAP Identity Management 8.0 SP02, globally unique translation keys are used within one and the same identity store. This means that if you define translations for a certain attribute, entry type or form once, then you can reuse the defined translation key (`#MX_`) in other places.

In the Identity Management Developer Studio, you can change the display language of attributes, entry types and forms in the Identity Management User Interface.

Note

Make sure your browser language settings correspond to the display language of the attributes, entry types and forms in the Identity Management User Interface.

Attributes

Context

To change the display language of attributes, do the following:

Procedure

1. In the tree view of the Identity Management Developer Studio, select the *Identity Store Schema*.
2. Select *Attributes* and choose *Open* from the context menu.
3. Select an attribute and choose *Open* from the context menu.
4. Select the *Presentation* tab and choose ... button for the *Display Name* of the attribute.
5. In the *Localize Text* dialog box, select the language of the attribute to be displayed in the Identity Management User Interface and choose *OK*.
6. Choose *Save*.

Entry Types

Context

To change the display language of entry types, do the following:

Procedure

1. In the tree view of the Identity Management Developer Studio, select the *Identity Store Schema*.
2. Select *Entry Types* and choose *Open* from the context menu.
3. Select an entry type and choose *Open* from the context menu.
4. Select the *General* tab and choose ... button for the *Display Name* of the entry type.
5. In the *Localize Text* dialog box, select the language of the entry type to be displayed in the Identity Management User Interface and choose *OK*.

6. Choose [Save](#).

Forms

Prerequisites

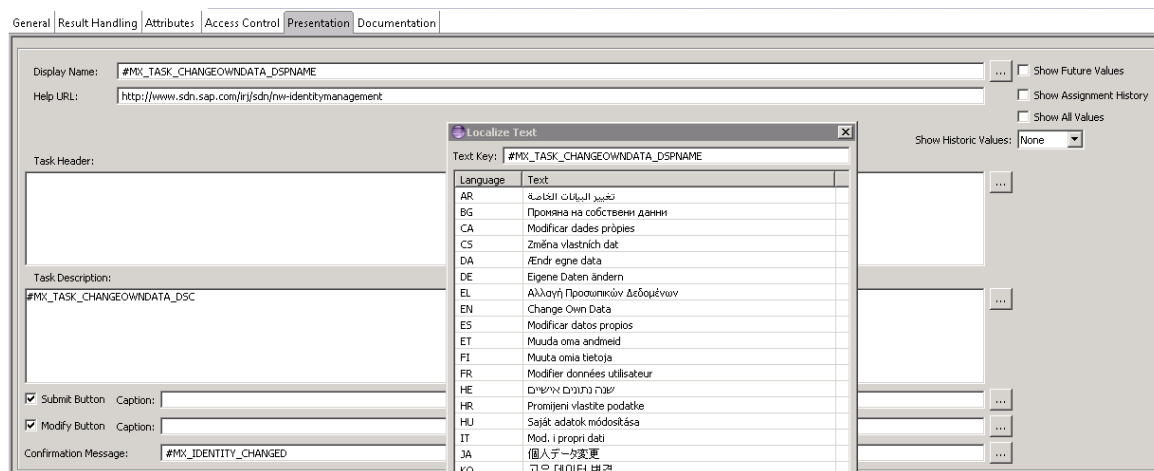
- You have checked out the forms package.

Context

To change the display language of forms, do the following:

Procedure

1. In the tree view of the Identity Management Developer Studio, select [Packages](#) folder and then select the forms package. For example, [com.sap.idm.forms.default](#).
2. In the [Forms](#) folder, select a form and choose [Properties](#) from the context menu.
3. Select the [Presentation](#) tab and choose ... button for the [Display Name](#) of the form.
4. In the [Localize Text](#) dialog box, select the language of the form to be displayed in the Identity Management User Interface and choose [OK](#).



5. Choose [Save](#).

3.2 Provisioning Framework for SAP Identity Management 8.0

The Provisioning framework for SAP Identity Management 8.0 provides the set of templates to use to connect SAP systems to SAP Identity Management and to set up the jobs and processes for provisioning the corresponding users and the corresponding assignments.

The Provisioning framework for SAP Identity Management 8.0 is distributed in several packages:

- **Engine** package - The engine package contains the core provisioning flow responsible for triggering the necessary processes (Provision, Deprovision and Modify) as well as common scripts used by all packages.
- **Connector** packages - The package for each connector contains the specific processes for provisioning to a specific system (for example ABAP).
- **Forms** package - The forms package contains the definition of all User Interface tasks for CRUD operations (create, read, update, delete) on different entry types.
- **Notification** package - The notification package contains the notification task and the notification templates used to send notifications from the SAP Provisioning framework, approval and attestation tasks.
- **Custom** package - The custom package allows customizations of the provisioning framework without changing other packages. It contains customizations and extensions made by the customers, default constants as well as custom scripts that could be changed later.

For more information about the package concept, see *Packages* in the *Related Information* section.

📌 Note

To perform successful provisioning, you need to import an engine package, a custom package and at least one connector package.

Related Information

[Packages \[page 91\]](#)

[Package Files \[page 399\]](#)

[Package Types \[page 404\]](#)

[Package Dependencies \[page 404\]](#)

[Package Content \[page 405\]](#)

[Updating the Provisioning Framework \[page 491\]](#)

[Importing the Provisioning Framework for SAP Identity Management 8.0 \[page 487\]](#)

3.2.1 Package Files

The tables below list the package files provided with the SAP Identity Management 8.0 frameworks.

Note

If you have installed SAP Identity Management with the Software Provisioning Manager 1.0 installation tool, to locate the package files, navigate to the directory where you have unpacked the Core component (ICCORE.SAR). For example: <Unpack_Directory>\ConfigurationPackages\SAP Provisioning Framework. See [Unpacking the Configuration Packages](#).

SAP Provisioning Framework

To locate the package files provided with this framework, navigate to <Install_folder>\Core\ConfigurationPackages\SAP Provisioning Framework.

File	Description
<code>com.sap.idm.connector.abap.businesssuite.idmpck</code>	The file of the ABAP BusinessSuite connector package com.sap.idm.connector.abap.businesssuite. This package contains the SAP provisioning framework connector for SAP Business Suite systems, including a connector-specific repository type, processes, jobs, and scripts.
<code>com.sap.idm.connector.abap.idmpck</code>	The file of the ABAP connector package com.sap.idm.connector.abap. This package contains the SAP provisioning framework connector for AS ABAP systems, including a set of connector-specific repository types, processes, jobs, and scripts.
<code>com.sap.idm.connector.ad.idmpck</code>	The file of the AD connector package com.sap.idm.connector.ad. This package contains the SAP provisioning framework connector for Microsoft Active Directory Server, including a connector-specific repository type, processes, jobs, and scripts.
<code>com.sap.idm.connector.asjava.idmpck</code>	The file of the AS Java connector package com.sap.idm.connector.asjava. This package contains the SAP provisioning framework connector for SAP NetWeaver AS for Java systems, including a set of connector-specific repository types, processes, jobs, and scripts.
<code>com.sap.idm.connector.custom.idmpck</code>	The file of the custom package com.sap.idm.connector.custom. This package allows customizations of the provisioning framework without changing other packages. It contains customizations and extensions made by the customer, default constants, and custom scripts that could be changed later.

File	Description
<code>com.sap.idm.connector.dualstack.idmpck</code>	The file of the dual-stack connector package <i>com.sap.idm.connector.dualstack</i>. This package contains the SAP Provisioning Framework connector for dual-stack systems (AS ABAP/AS Java), including a set of connector-specific repository types and processes.
<code>com.sap.idm.connector.hana.idmpck</code>	The file of the HANA connector package <i>com.sap.idm.connector.hana</i>. This package contains the SAP provisioning framework connector for SAP HANA, including a connector-specific repository type, processes, jobs, and scripts.
<code>com.sap.idm.connector.sfsf.idmpck</code>	The file of the SuccessFactors connector package <i>com.sap.idm.connector.sfsf</i>. This package contains the SAP provisioning framework connector for SuccessFactors, including a connector-specific repository type, processes, jobs, and scripts.
<code>com.sap.idm.connector.sun.idmpck</code>	The file of the Sun connector package <i>com.sap.idm.connector.sun</i>. This package contains the SAP provisioning framework connector for SUN LDAP Server, including a connector-specific repository type, processes, and jobs.
<code>com.sap.idm.provisioning.engine.idmpck</code>	The file of the engine package <i>com.sap.idm.provisioning.engine</i>. This package contains the provisioning engine/core logic of the provisioning framework, including the provisioning processes, utility processes, and scripts used by the engine itself or the connectors.
<code>com.sap.idm.connector.sci.idmpck</code>	The file of the SAP Cloud Identity connector package <i>com.sap.idm.connector.sci</i>. This package contains the SAP provisioning framework connector for SAP Cloud Identity service, including a connector specific repository type, processes, jobs and scripts
<code>com.sap.idm.connector.scim.idmpck</code>	The file of the SCIM connector package <i>com.sap.idm.connector.scim</i>. This package contains the SAP Provisioning Framework connector for SCIM 2.0 based systems, used for identity provisioning. It includes a specific repository type and a set of processes, jobs and scripts.

GRC Provisioning Framework

To locate the package file provided with this framework, navigate to
<Install_folder>\Core\ConfigurationPackages\GRC Provisioning Framework.

File	Description
com.sap.idm.grc.grc10.idmpck	The file of the Governance, Risk and Compliance (GRC) provisioning framework package com.sap.idm.grc.grc10. This package contains the GRC 10.0 Provisioning Framework, including the repository type, provisioning processes, utility processes, initial load jobs and scripts.

HCM Integration Framework

To locate the package file provided with this framework, navigate to
<Install_folder>\Core\ConfigurationPackages\HCM.

File	Description
com.sap.idm.hcm.idmpck	The file of the HCM integration package com.sap.idm.hcm. This package contains the processes, constants, and scripts necessary for the SAP HCM integration.

Lotus Notes

To locate the package file provided with this framework, navigate to
<Install_folder>\Core\ConfigurationPackages\Lotus Notes.

File	Description
com.sap.idm.connector.notes.idmpck	The file of the Lotus Notes connector package com.sap.idm.connector.notes. This package contains the SAP Provisioning Framework connector for IBM Lotus Notes systems, including a set of connector specific repository type, processes, jobs and scripts.

Notification Framework

To locate the package file provided with this framework, navigate to
<Install_folder>\Core\ConfigurationPackages\Notification.

File	Description
<code>com.sap.idm.util.notification.idmpck</code>	The file of the notification package com.sap.idm.util.notification. This package contains the notification framework, including the message templates, a process for sending e-mail notifications, and a set of notification constants.

Forms

To locate the package files provided with this framework, navigate to
<Install_folder>Core\ConfigurationPackages\Forms.

File	Description
<code>com.sap.idm.forms.default.idmpck</code>	The file of the forms package com.sap.idm.forms.default. This package contains the definition of all user interface tasks for CRUD operations (create, read, update, delete) on different entry types as well as a set of WebDynpro-based sample display and search forms for the entry types, a few processes, package constants, and one script.
<code>com.sap.idm.forms.html5.idmpck</code>	The file of the HTML5 forms package com.sap.idm.forms.html5. This package contains a set of SAP UI5-based sample forms for displaying different entry types, and forms for managing user attributes.

Reporting

To locate the package file provided with this framework, navigate to
<Install_folder>\Core\ConfigurationPackages\Reporting.

File	Description
<code>com.sap.idm.reporting.idmpck</code>	The file of the reporting package com.sap.idm.reporting. This package contains regular processes and forms required for configuring standard reports.

3.2.2 Package Types

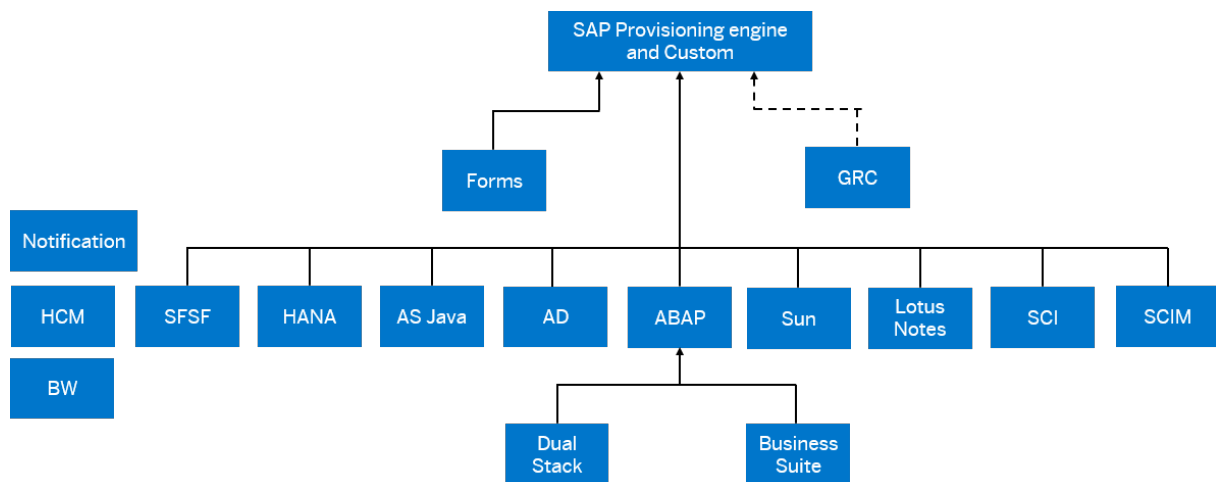
The following package types are defined:

Package Types	Description
SAP package	SAP read-only package A package distributed by SAP that cannot be modified by customers. For example, the engine package.
	SAP modifiable package A package distributed by SAP that can be modified by customers. For example, a package for a specific connector, such as com.sap.idm.connector.hana. When a package of this type is modified, its type changes to SAP package modified by customer.
	SAP package modified by customer A package distributed by SAP that used to be an SAP modifiable package that was later modified by customers.
Customer package	A package created by customers.

3.2.3 Package Dependencies

There are dependencies between packages that require some of the packages to be imported in advance.

The figure below shows the package dependency structure.



Most of the packages depend on one or more packages. For example, the Business Suite connector package depends on the ABAP connector package, the engine package and the custom package. You therefore need to import these in the following sequence:

1. Import the engine and the custom package.
2. Import the ABAP connector package.
3. Import the Business Suite connector package.

📌 Note

It is important to import packages in the correct sequence. Otherwise, the corresponding references will not be resolved.

The GRC Provisioning framework package does not have any technical dependency to the engine package. It can be imported on its own, although it cannot function properly without the engine package.

3.2.4 Package Content

A package is a collection of configuration information, such as constants, scripts, repository types, processes, forms and jobs.

Related Information

[Engine Package \[page 406\]](#)

[ABAP Connector Package \[page 438\]](#)

[ABAP BusinessSuite Connector Package \[page 442\]](#)

[AD Connector Package \[page 445\]](#)

[AS Java Connector Package \[page 448\]](#)

[Custom Package \[page 472\]](#)

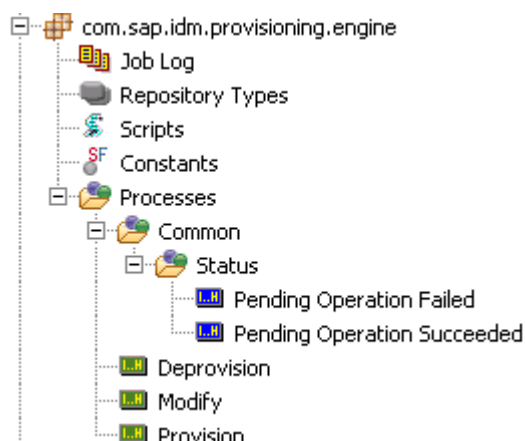
[Dual Stack Connector Package \[page 451\]](#)

[HANA Connector Package \[page 454\]](#)
[Sun Connector Package \[page 465\]](#)
[Success Factors Connector Package \[page 457\]](#)
[Lotus Notes Connector Package \[page 468\]](#)
[Forms Package \[page 473\]](#)
[HTML5 Forms Package \[page 475\]](#)
[Notification Package \[page 477\]](#)
[HCM Integration Package \[page 484\]](#)

3.2.4.1 Engine Package

The engine package *com.sap.idm.provisioning.engine* contains the core provisioning flow that is responsible for triggering the necessary processes (Provision, Deprovision and Modify), as well as common processes and common scripts used by all packages.

The structure of the engine package is as follows:



The following package constants are defined:

Package Constant	Value	Description
SAP_DELIMITER	Default value:	Separates multivalue attributes. To make the SAP_DELIMITER package constant available for jobs in other packages, you need to add link to the <code>getMultiValueDelimiter()</code> package script that references the package constant using the following syntax <code>%\$pck.SAP_DELIMITER%</code>.

Package Constant	Value	Description
PRIVI- LEGES_DEAS- SIGNMENT_DE- LAY	Default value in seconds: 60	Delays the removal of user's account privilege from a repository. Ensures that all user's privileges are removed from a repository before the account privilege. If you need to increase the delay, log on to SAP Identity Management Administration User Interface and navigate to System Configuration > Packages > com.sap.idm.provisioning.engine > Constants.

Core Provisioning Processes

There are three core provisioning processes in the framework:

- Deprovision - Deprovision process that performs the deprovisioning from the connected backend system(s).
- Modify - Modify process that performs the modifications in the connected backend system(s).
- Provision - Provision process that performs the provisioning to the connected backend system(s).

These processes contain the logic to decide which plug-in processes (contained in the connectors) are called. For more information about the logic behind these processes and their execution process, see: *Process Execution Description for Core Provisioning Processes*.

Common Processes

The common processes include status processes used by the core provisioning processes Provision and Deprovision:

- Pending Operation Failed
- Pending Operation Succeeded

For more information about how the core provisioning processes Provision and Deprovision use these status processes, see: *Provision* and *Deprovision*.

Along with the processes, the engine package contains a number of scripts used in various cases, such as encrypting passwords, generating display names, getting an audit ID.

Related Information

[Process Execution Description for Core Provisioning Processes \[page 408\]](#)

[Provision \[page 408\]](#)

[Deprovision \[page 421\]](#)

3.2.4.1.1 Process Execution Description for Core Provisioning Processes

The core part of the provisioning framework contains the central logic. This logic is implemented in the core provisioning processes. These decide which plug-in processes are called, while the connectors contain the plug-in tasks that are referenced from the repositories.

This section describes the execution process of the core provisioning processes in SAP Provisioning Framework, together with the logic behind these processes. The framework contains three core provisioning processes:

- Provision
- Deprovision
- Modify

Related Information

[Provision \[page 408\]](#)

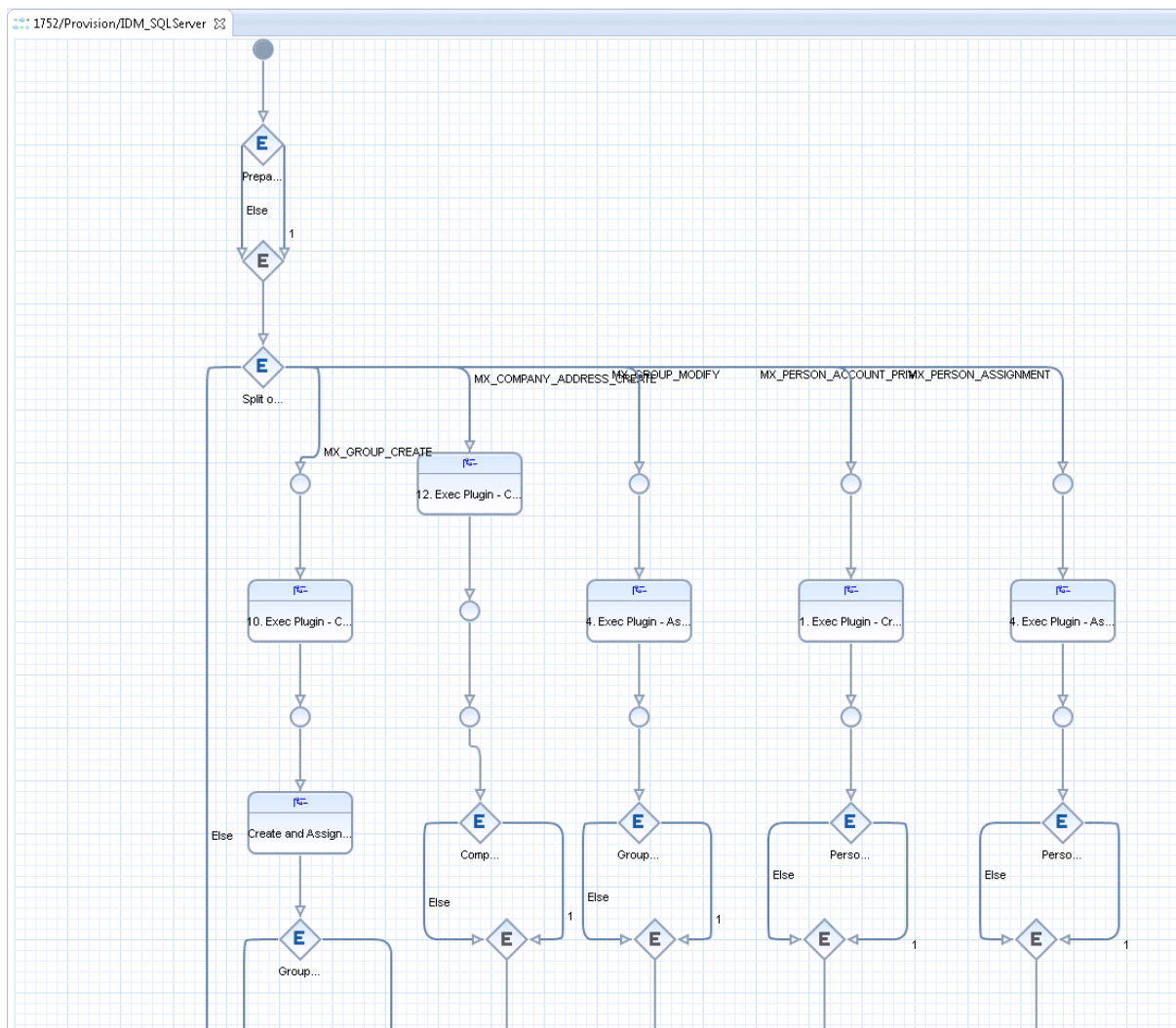
[Deprovision \[page 421\]](#)

[Modify \[page 435\]](#)

3.2.4.1.1.1 Provision

When privileges are assigned, Identity Management performs the defined "add member event" processes. The Provision process performs the provisioning to the connected back end system(s).

The Provision process is an Add Member process:



The process consists of two internal execution tasks:

- Prepare Provisioning
- Split on EntryType Operation

It also configures specific chain result actions (on the Result Handling tab in the process properties):

Chain Result Actions	
Execute Process on OK Result:	1729/Pending Operation Succeeded
Execute Process on Failed Result:	1738/Pending Operation Failed

Each task in the Provision process is given a status after being processed. If the tasks are executed successfully, the entire process is given the status Pending Operation Succeeded. This is an internal execution

process that uses the Java library method `com.sap.idm.pf.common.ExtACGroupSucceeded.exec` to set the status of the entire provision process to "0" (pending operation applied).

If any of the tasks in the chain result in error (failed or rejected), the status process Pending Operation Failed is set. This is an internal execution process that uses the Java library method `com.sap.idm.pf.common.ExtACGroupFailed.exec` to set the status of the entire provision process to "3" (pending operation not applied).

Both status tasks are located under `Processes\Common\Status` in the tree structure of the package.

3.2.4.1.1.1.1 Prepare Provisioning

This task (internal execution task) uses the Java library method `com.sap.idm.pf.prov.ExtPrepareProvision.exec` to do the following:

- Check if the repositories are marked for synchronization (which is the expected behavior for all repositories in the framework). If this is the case, proceed with provisioning. If not, set the task status to error and abort the provisioning process.
- Make sure that the links are defined by attribute `MXREF_...`, both when adding a privilege/role to a user and when adding a user to a privilege/role, giving the same pending value object and referenced object in both cases.
- Extract all relevant information from the pending value object group.
- Store this information in context variables that are used by tasks in a subsequent processing phase.

The following is a list of context variables that might be of interest:

Context Variable Name	Description
PENDINGMSKEY	The MSKEY of the pending object (the leading object in the group).
PENDINGTARGETMSKEY	The MSKEY of the entry that the privilege(s) is/are assigned to.
PENDINGPRIVILEGEREPNAME	The repository defined for the privilege.
PENDINGPRIVILEGEMSKEY	The MSKEY of the privilege(s) assigned.
GROUP_PENDINGMSKEYS	Pipe () separated list of all pending value objects (PVOs) in the group, listed by their MSKEYs. Example: 123 124 125
GROUP_PENDINGMSKEYS_PRIVILEGEMSKEYS	List of PVO-privilege pairs listed by their MSKEYs (for each PVO listed by the context variable <code>GROUP_PENDINGMSKEYS</code>).

Context Variable Name	Description
	Example:
	123 6871!!124 6872!!125 6873

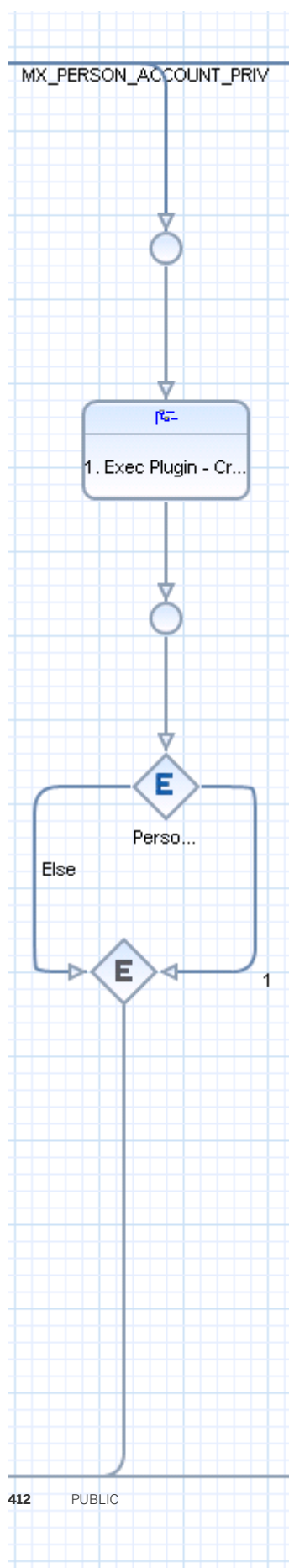
3.2.4.1.1.1.2 Split on Entry Type Operation

This task (internal execution task), is a switch task that uses the Java library method `com.sap.idm.pf.prov.ExtSplitObjTypeProv.exec` to determine which operation should be executed for each of the entry types `MX_PERSON`, `MX_GROUP` and `MX_COMPANY_ADDRESS`:

- For `MX_PERSON`, the task checks whether the privilege to be assigned to the user is an account privilege
 - If this is the case, proceed with the steps under case node [MX_PERSON_ACCOUNT_PRIV](#).
 - If this is not the case, proceed with the steps under case node [MX_PERSON_ASSIGNMENT](#).
- For `MX_GROUP`, the task checks whether group provisioning is allowed. If it is not allowed, set the task status to Error and abort provisioning. If group provisioning is allowed, continue by checking whether the group already exists:
 - If the group exists, run the steps under case node [MX_GROUP_MODIFY](#).
 - If not, run the steps under case node [MX_GROUP_CREATE](#).
- For `MX_COMPANY_ADDRESS`, the task checks whether the company address already exists:
 - If the company address exists, you do not need to do anything.
 - If the company address does not exist, proceed with the steps defined under case node [MX_COMPANY_ADDRESS_CREATE](#)

See the subsections below for details about the steps defined for each case node.

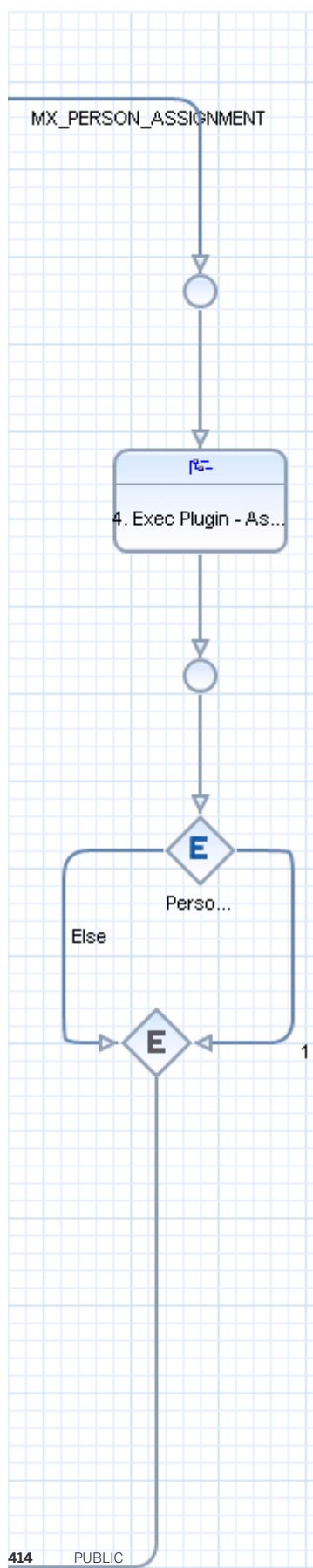
MX_PERSON_ACCOUNT_PRIV



This case node contains three tasks:

Task Name	Description
<i>1. Exec Plugin - Create User</i>	This task executes the plug-in task <i>1. Create User</i> defined on the repository.
<i>Wait on previous result and trigger error (if any)</i>	The purpose of this task is to make sure that the system waits until the user creation process started by the plug-in task is completed before moving to the next step.
<i>Person after Creation</i>	This task uses the Java library method <code>com.sap.idm.pf.prov.ExtProvPersonAccountEND.exec</code> to: • Add system privilege to the user account. • If the assignment is successful, sends a notification that the user account now is created (if notifications are implemented).

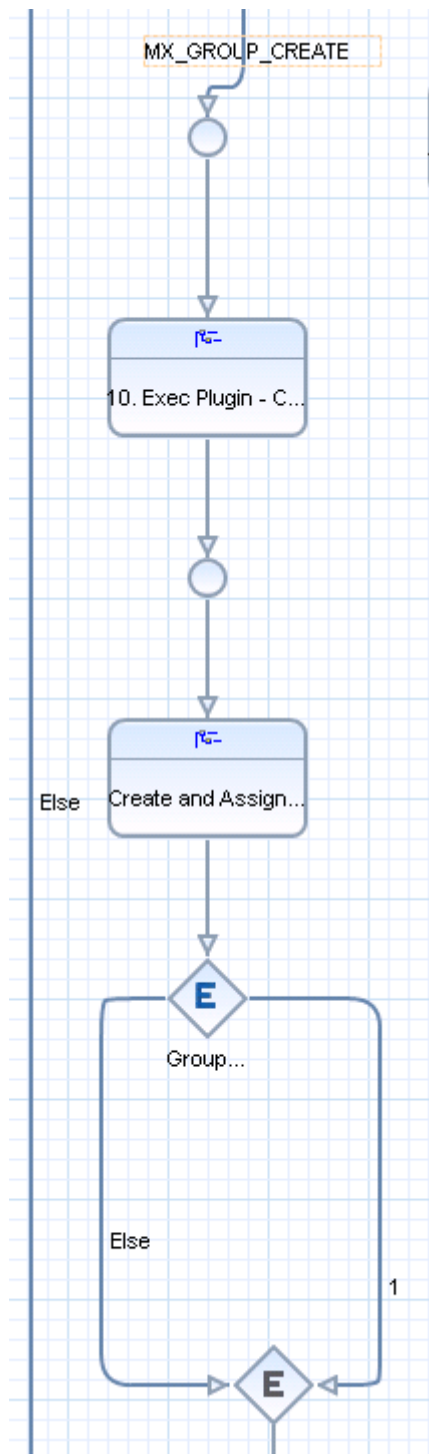
MX_PERSON_ASSIGNMENT



This case node contains three tasks:

Task Name	Description
<i>4. Exec Plugin - Assign User Membership</i>	This task executes the plug-in task <i>4. Assign User Membership</i> defined on the repository.
<i>Wait on previous result and trigger error (if any)</i>	The purpose of this task is to make sure that the system waits until the privilege/membership assignment process started by the plug-in task is completed before moving to the next step.
<i>Person after Assignment Provisioning</i>	This task uses the Java library method <code>com.sap.idm.pf.prov.ExtProvPersonAssignments</code> <code>END.exec</code> to: • Check if the privilege to be assigned is a group privilege, and if so make sure that the group membership is updated. • If the assignment is successful, send a notification that the user now is granted the requested membership(s) (if notifications are implemented).

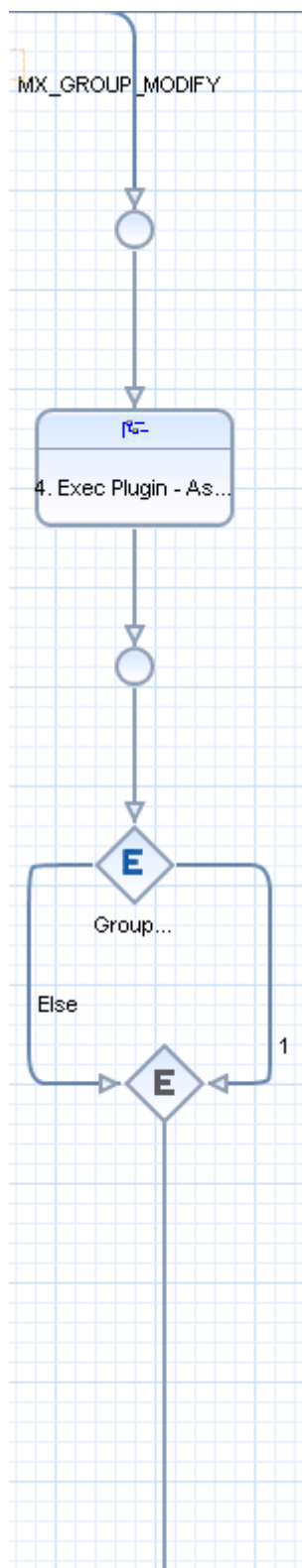
MX_GROUP_CREATE



This case node contains three tasks:

Task Name	Description
<i>Exec Plugin - Create Group with Membership Assgnment</i>	This task executes the plug-in task <i>10. Create Group with Membership Assignment</i> defined on the repository.
<i>Create and Assign Group Privilege</i>	This task creates the group privilege in the following format: <pre>PRIV:GROUP:% \$rep.\$NAME%:\$FUNCTION.sap_core_escapeC omma(%ACCOUNT%\$rep.\$NAME%)\$\$</pre> where the global Java script <i>sap_core_escapeComma</i> is used to make sure that the group's MSKEYVALUE does not contain any commas (,) and %\$rep.\$Name% is the name of the repository the privilege belongs to. The task also assigns the created privilege to the group.
<i>Group after Creation</i>	This task uses the Java library method <code>com.sap.idm.pf.prov.ExtProvGroupCreateEND.execute</code> to: • Add system privilege to the group. • Update the group privilege triggers (MX_ADD/DEL_MEMBER_TASK, MX_VALIDATE_ADD/DEL_TASK). • Make sure for all group members that attribute MXREF_MX_PRIVILEGE holds a reference to the group privilege.

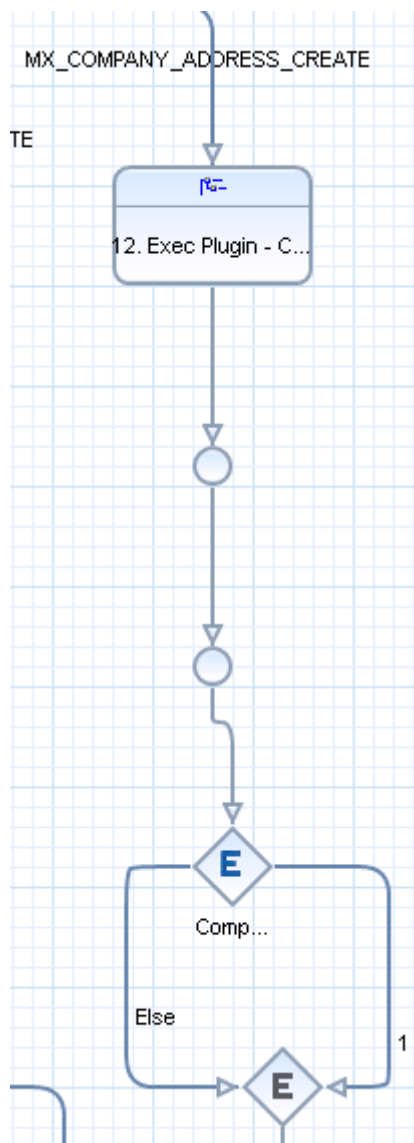
MX_GROUP_MODIFY



This case node contains three tasks:

Task Name	Description
<i>4. Exec Plugin - Assign User Membership</i>	This task executes the plug-in task <i>4. Assign User Membership</i> defined on the repository.
<i>Wait on previous result and trigger error (if any)</i>	The purpose of this task is to make sure that the system waits until the privilege/membership assignment process started by the plug-in task is completed before moving to the next step.
<i>Group after modify</i>	This task uses the Java library method <code>com.sap.idm.pf.prov.ExtProvGroupModifyEND.execute</code> to: • Check if the privilege to be assigned is a group privilege, and if so make sure that the group membership is updated. • Make sure for all group members that attribute <code>MXREF_MX_PRIVILEGE</code> holds a reference to the group privilege.

MX_COMPANY_ADDRESS_CREATE



This case node contains three tasks:

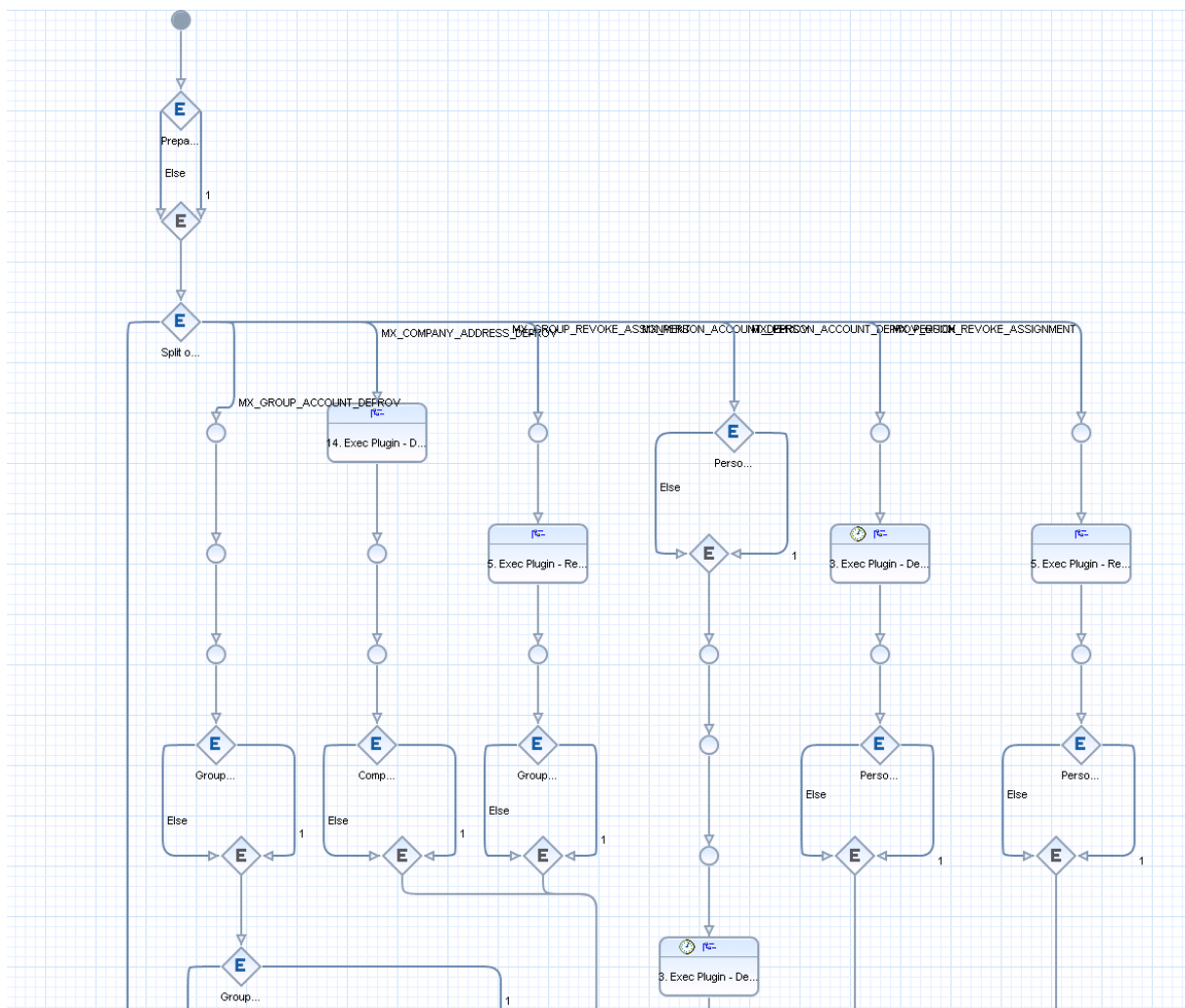
Task Name	Description
<i>12. Exec Plugin - Create Company Address</i>	This task executes the plug-in task <i>12. Create Company Address</i> defined on the repository.
<i>Wait on previous result and trigger error (if any)</i>	The purpose of this task is to make sure that the system waits until the company address creation process started by the plug-in task is completed before moving to the next step.

Task Name	Description
<i>Company Address after Creation</i>	This task uses the Java library method <code>com.sap.idm.pf.prov.ExtProvCompanyAddressCreateEND.exec</code> to add system privilege to the created company address entry.

3.2.4.1.1.2 Deprovision

When privileges are de-assigned, Identity Management performs the defined *remove member event* task. The *Deprovision* process performs the deprovisioning from the connected back end system(s).

The *Deprovision* process is a *Remove Member* process:



The task group consists of two internal execution tasks:

- Prepare Deprovisioning

- Split on Entry Type

It also configures specific chain result actions (on the Result Handling tab in the task properties):

Chain Result Actions

Execute Process on OK Result:	1729/Pending Operation Succeeded	...
Execute Process on Failed Result:	1738/Pending Operation Failed	...

Each task in the *Deprovision* process is given a status after being processed. If the tasks are executed successfully, the entire process is given the status *Pending Operation Succeeded*. This is an internal execution process that uses the Java library method `com.sap.idm.pf.common.ExtACGroupSucceeded.exec` to set the status of the entire deprovision process to "0" (pending operation applied).

If any of the tasks in the chain result in errors (failed or rejected), the status process *Pending Operation Failed* is set. This is an internal execution process that uses the Java library method `com.sap.idm.pf.common.ExtACGroupFailed.exec` to set the status of the entire deprovision process to "3" (pending operation not applied).

Both status processes are located under `Processes\Common\Status` in the tree structure of the package.

3.2.4.1.1.2.1 Prepare Deprovisioning

This task (internal execution task) uses the Java library method `com.sap.idm.pf.deprov.ExtPrepareDeprovision.exec` to do the following:

- Check if the repositories are marked for synchronization (the expected behavior for all repositories in the framework) . If this is the case, proceed with deprovisioning. If not, set the task status to error and abort the deprovisioning process.
- Make sure that the links are defined by attribute `MXREF_...`, both when removing a privilege/role from a user and when removing a user from a privilege/role, giving the same pending value object and referenced object in both cases.
- Extract all relevant information from the pending value object group.
- Store this information in context variables that are used by tasks during a subsequent processing phase.

The following is a list of context variables that might be useful:

Context Variable Name	Description
PENDINGMSKEY	The MSKEY of the pending object (the leading in the group).
PENDINGTARGETMSKEY	The MSKEY of the entry that the privilege(s) is /are assigned to.
PENDINGPRIVILEGEREPNAME	The repository defined for the privilege.

Context Variable Name	Description
PENDINGPRIVILEGEMSKY	The MSKEY of the privilege(s) assigned.
GROUP_PENDINGMSKEYS	Pipe () separated list of all pending value objects (PVOs) in the group, listed by their MSKEYs. Example: <pre>123 124 125</pre>
GROUP_PENDINGMSKEYS_PRIVILEGEMSKYS	List of PVO-privilege pairs listed by their MSKEYs (for each PVO listed by the context variable GROUP_PENDINGMSKEYS). Example: <pre>123 6871 ! ! 124 6872 ! ! 125 6873</pre>

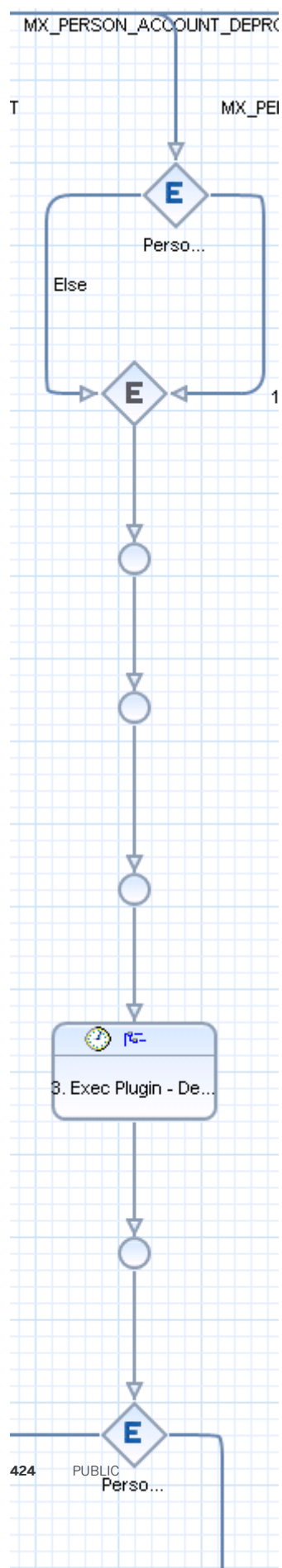
3.2.4.1.1.2.2 Split on Entry Type (Deprovisioning)

This task (internal execution task), is a switch task that uses the Java library method `com.sap.idm.pf.deprov.ExtSplitObjTypeDeprov.exec` to determine which operation should be executed for each of the entry types `MX_PERSON`, `MX_GROUP` and `MX_COMPANY_ADDRESS`:

- For `MX_PERSON`, the task checks whether the privilege to be revoked from the user is an account privilege:
 - If this is the case, proceed with the steps under case node [MX_PERSON_ACCOUNT_DEPROV](#) or [MX_PERSON_ACCOUNT_DEPROV_QUICK](#). The node to be selected depends on which repository constant is set. If repository constant `MX_ALLOWS_QUICK_ENTRY_DEL` is set, the tasks in case node [MX_PERSON_ACCOUNT_DEPROV_QUICK](#) are executed. Otherwise, proceed with case node [MX_PERSON_ACCOUNT_DEPROV](#).
 - If the privilege is not an account privilege, proceed with the steps under case node [MX_PERSON_REVOKE_ASSIGNMENT](#).
- For `MX_GROUP`, the task checks the state of the group validating whether the group is already being deleted. The deprovisioning process will only proceed if the group is not already being deleted. The task will continue with a check regardless of whether the privilege that is about to be revoked is an account privilege or not:
 - If the privilege is an account privilege, run the steps under case node [MX_GROUP_ACCOUNT_DEPROV](#).
 - If not, run the steps under case node [MX_GROUP_REVOKE_ASSIGNMENT](#).
- For `MX_COMPANY_ADDRESS`, proceed with the steps defined under case node [MX_COMPANY_ADDRESS_DEPROV](#).

See subsections below for details about the steps defined for each case node.

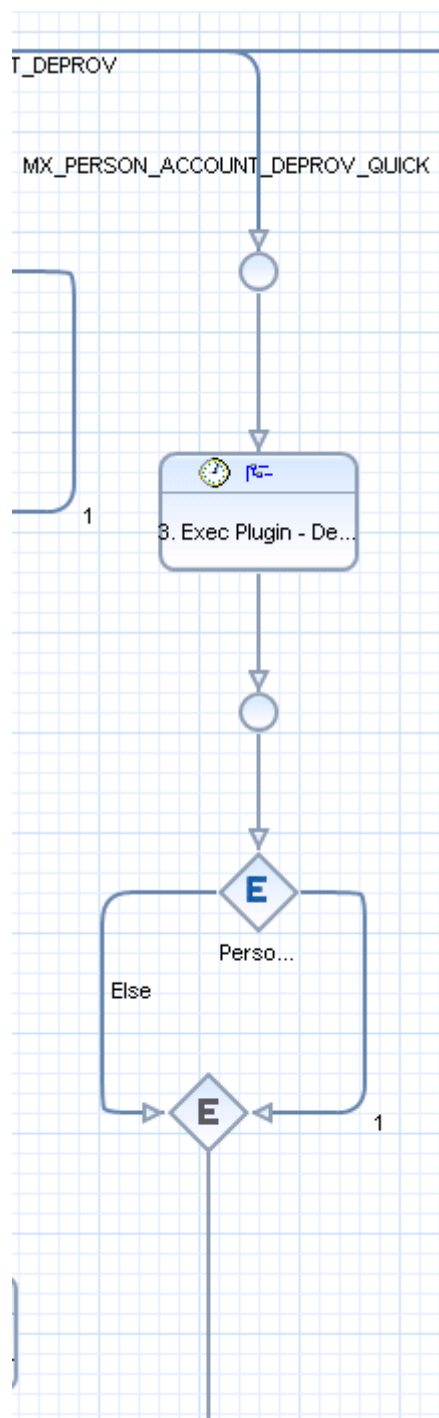
MX_PERSON_ACCOUNT_DEPROV



This case node contains five tasks:

Task Name	Description
<i>Person before Deprovision</i>	This task uses the Java library method <code>com.sap.idm.pf.deprov.ExtDeprovPersonDeprovisionSTART.exec</code> to check for any non-account privileges for this user account. If any are found, start the deassignment process to revoke membership of all non-account privileges for the user. If not, proceed to the next task.
<i>Wait for all Non-Account Privs Deassignment</i>	The purpose of this task is to make sure that the system waits until all non-account privileges are deassigned, and the process started by the previous task is completed before moving to the next step.
<i>3. Exec Plugin - Delete User</i>	This task executes the plug-in task <i>3. Delete User</i> defined on the repository.
<i>Wait on previous result and trigger error (if any)</i>	The purpose of this task is to make sure that the system waits until the user deletion process started by the plug-in task is completed before moving to the next step.
<i>Person after Deprovision</i>	This task uses the Java library method <code>com.sap.idm.pf.deprov.ExtDeprovPersonDeprovisionEND.exec</code> to: • Remove the system privilege. • Remove attribute ACCOUNT (ACCOUNT%\$rep.\$NAME%) for the user. • If deprovisioning is successful, send notification that the user has been deleted/deprovisioned from the system (if notifications are implemented).

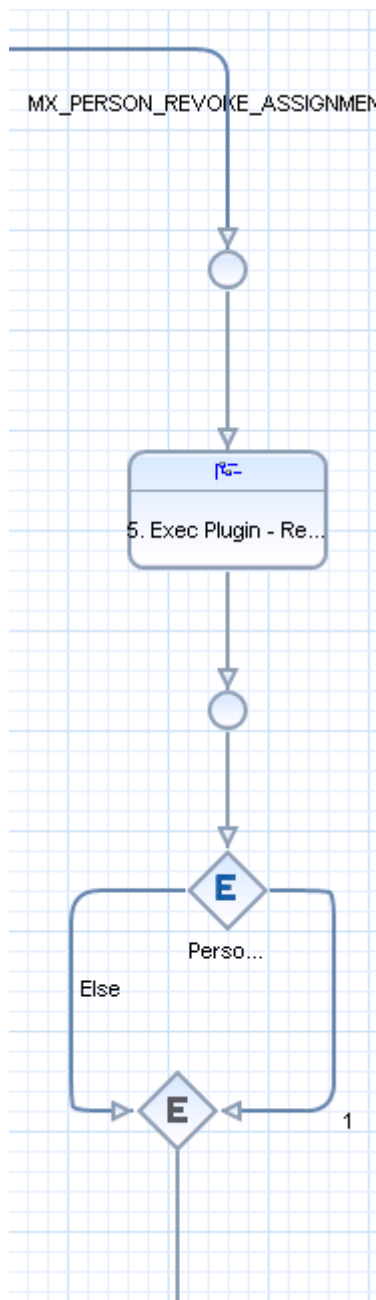
MX_PERSON_ACCOUNT_DEPROV_QUICK



This case node is similar to case node [MX_PERSON_ACCOUNT_DEPROV](#), except for the checking for and deassigning of any non-account privileges for the user account. Instead, the tasks under this case node proceed directly with deletion of the user account. The case node contains three tasks:

Task Name	Description
3. Exec Plugin - Delete User	This task executes the plug-in task 3. Delete User defined on the repository.
Wait on previous result and trigger error (if any)	The purpose of this task is to make sure that the system waits until the user deletion process started by the plug-in task is completed before moving to the next step.
Person after Quick Deprovision	This task uses the Java library method <code>com.sap.idm.pf.deprov.ExtDeprovPersonDeprovisionQuickEND.exec</code> to: • Remove the system privilege. • Remove attribute ACCOUNT (ACCOUNT%\$rep.\$NAME%) for the user. • If deprovisioning is successful, send notification that the user has been deleted/deprovisioned from the system (if notifications are implemented).

MX_PERSON_REVOKE_ASSIGNMENT

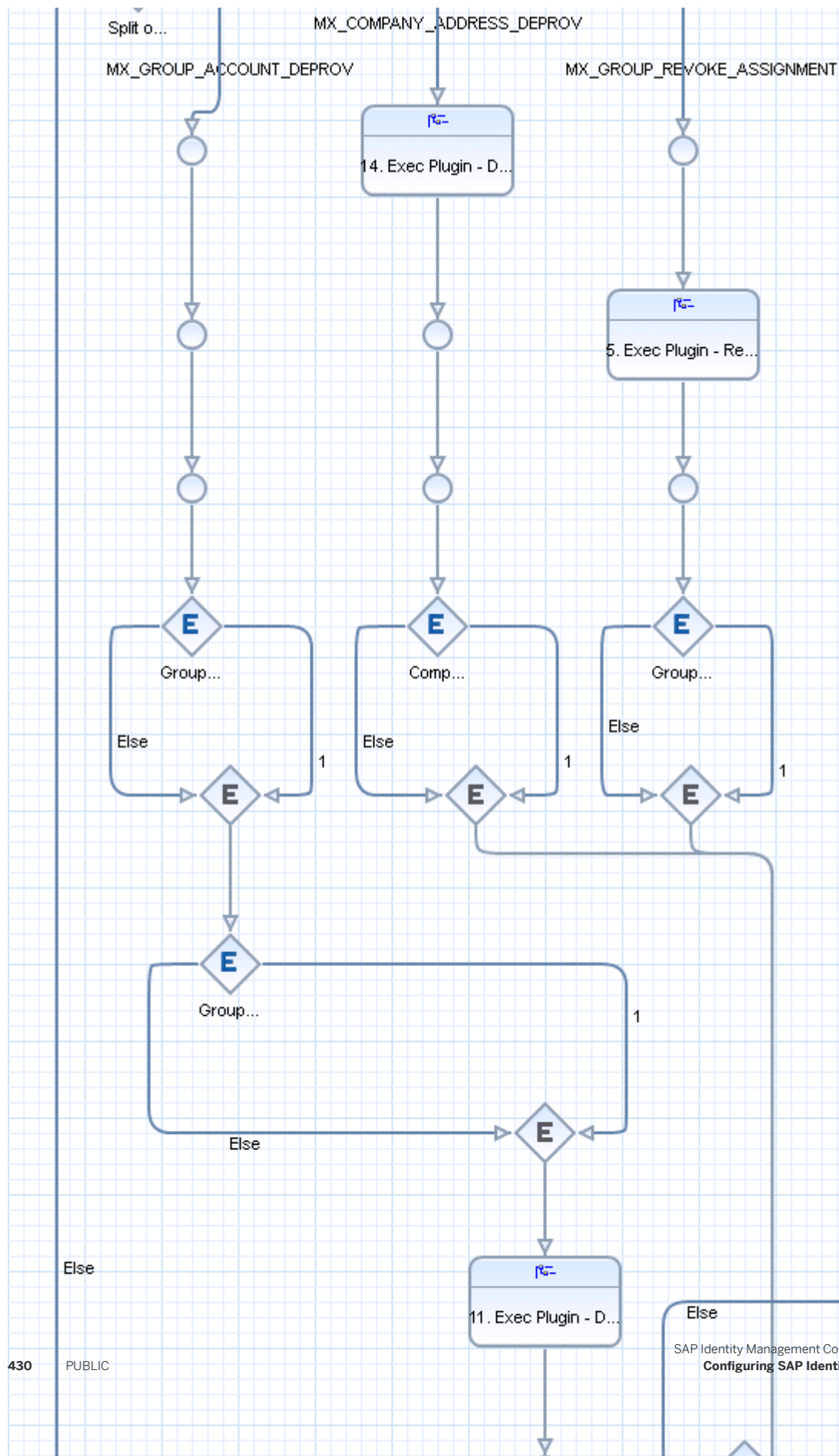


This case node contains three tasks:

Task Name	Description
5. Exec Plugin - Revoke User Membership	This task executes the plug-in task 5. Revoke User Membership defined on the repository.

Task Name	Description
<i>Wait on previous result and trigger error (if any)</i>	The purpose of this task is to make sure that the system waits until the privilege/membership revocation process started by the plug-in task is completed before moving to the next step.
<i>Person after Assignment Revocation</i>	This task uses the Java library method <code>com.sap.idm.pf.deprov.ExtDeprovPersonAssignmentRevocationEND.exec</code> to: • Check if the privilege to be revoked is a group privilege. If this is the case, make sure that the group membership is updated accordingly. • If the revocation is successful, send notification that the membership has been revoked from the user (if notifications are implemented).

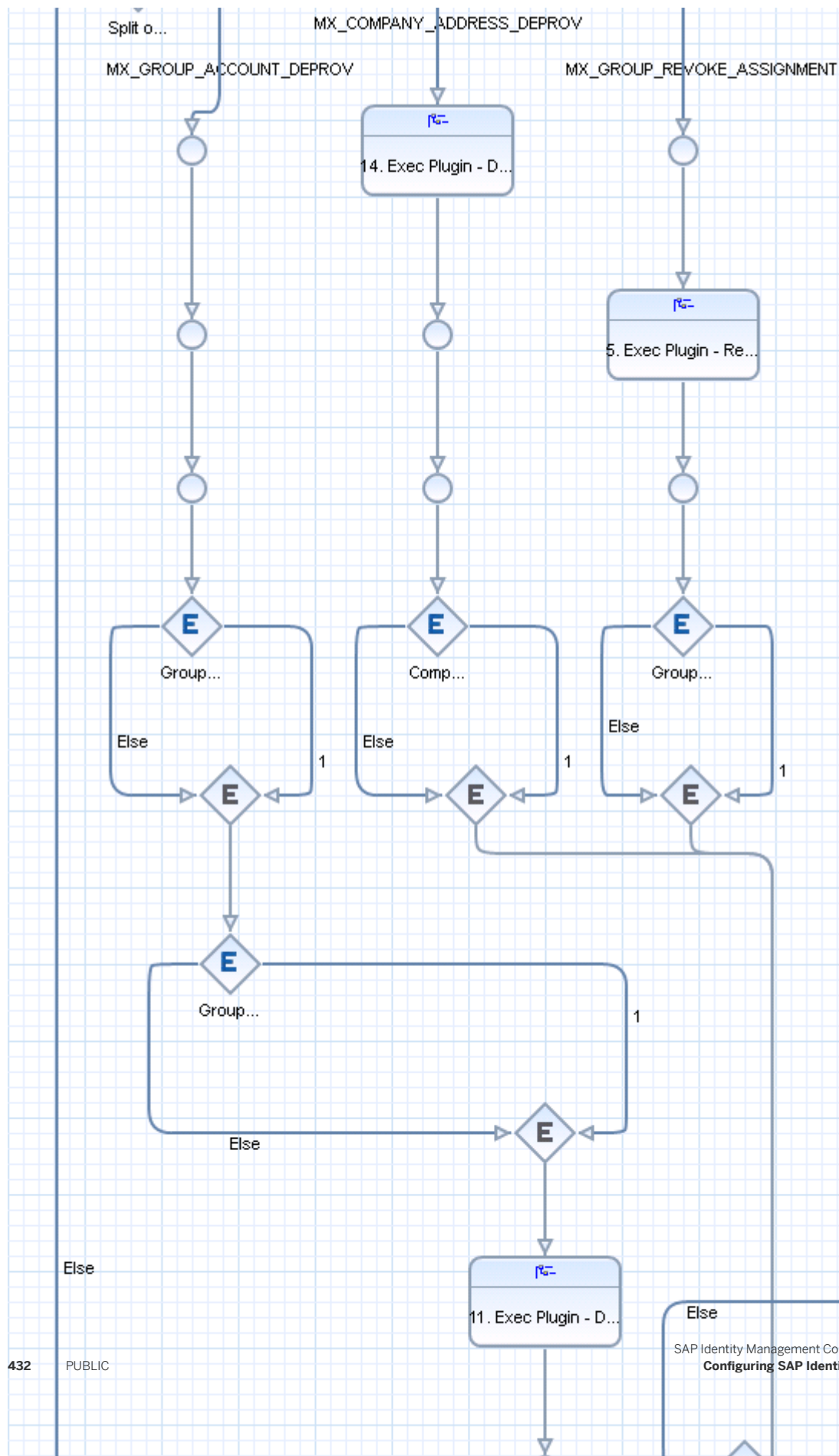
MX_GROUP_ACCOUNT_DEPROV



This case node contains five tasks:

Task Name	Description
<i>Group - prepare & updateMembers - With WAIT After</i>	This task uses the Java library method <code>com.sap.idm.pf.deprov.ExtDeprovGroupDeprovisionSTART.exec</code> to obtain a list of all members of the group account and remove group privilege references.
<i>Group - remove Assignments - With WAIT After</i>	This task uses the Java library method <code>com.sap.idm.pf.deprov.ExtDeprovGroupDeprovisionSTART2.exec</code> to update the privilege references of the removed group's parents.
<i>11. Exec Plugin - Delete container object - waitafter</i>	This task executes the plug-in task 11. Delete group defined on the repository.
<i>Wait on previous result and trigger error (if any)</i>	The purpose of this task is to make sure that the system waits until the group object deletion process started by the plug-in task is completed before moving to the next step.
<i>Group - After Account Deletion</i>	This task uses the Java library method <code>com.sap.idm.pf.deprov.ExtDeprovGroupDeprovisionEND.exec</code> to: • Revoke the system privilege for the group. • Remove the group privilege. • Remove the ACCOUNT attribute (ACCOUNT%\$rep.\$NAME%) for the group.

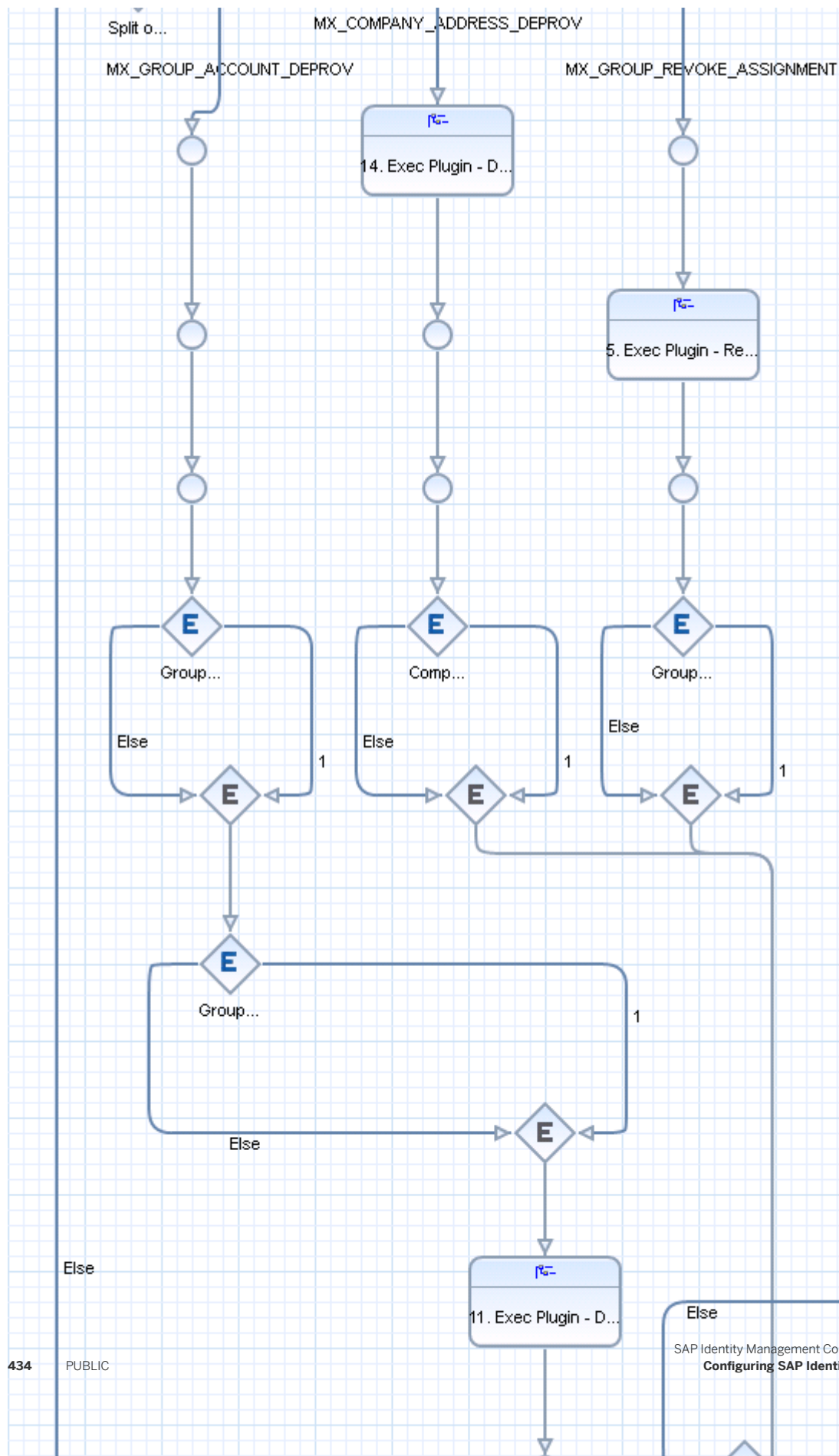
MX_GROUP_REVOKE_ASSIGNMENT



This case node contains three tasks:

Task Name	Description
<i>5. Exec Plugin - Revoke User Membership</i>	This task executes the plug-in task <i>5. Revoke User Membership</i> defined on the repository.
<i>Wait on previous result and trigger error (if any)</i>	The purpose of this task is to make sure that the system waits until the privilege/membership assignment process started by the plug-in task is completed before moving to the next step.
<i>Group - After Membership Revocation</i>	This task uses the Java library method <code>com.sap.idm.pf.deprov.ExtDeprovGroupMembershipRevocationEND.exec</code> to make sure that the group references for all revoked privileges are also revoked.

MX_COMPANY_ADDRESS_DEPROV



This case node contains three tasks:

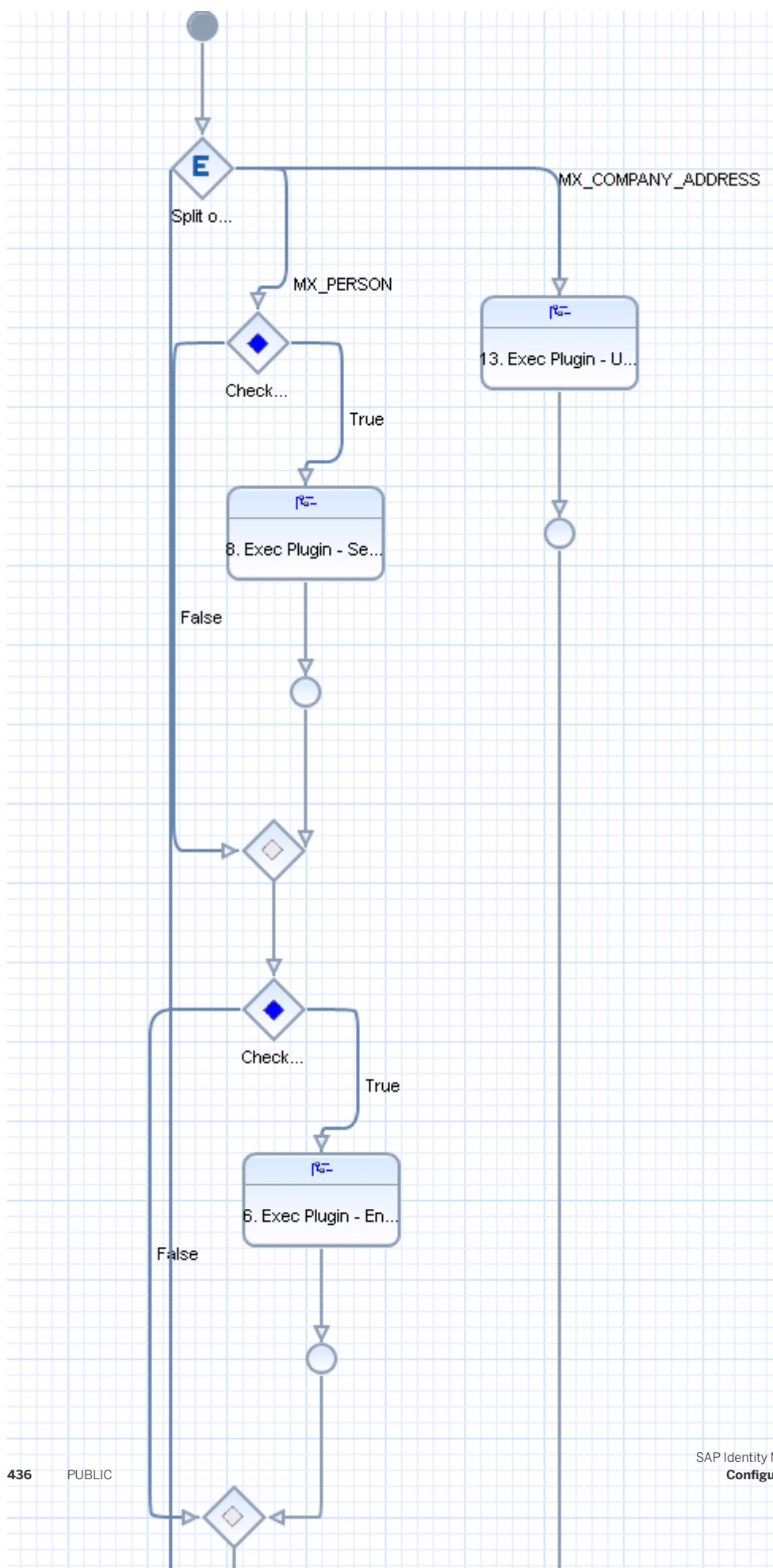
Task Name	Description
<i>14. Exec Plugin - Delete Company Address</i>	This task executes the plug-in task <i>14. Delete Company Address</i> defined on the repository.
<i>Wait on previous result and trigger error (if any)</i>	The purpose of this task is to make sure that the system waits until the company address creation process started by the plug-in task is completed before moving to the next step.
<i>Company Address after Deprovision</i>	This task uses the Java library method <code>com.sap.idm.pf.deprov.ExtDeprovCompanyAddressDeprovisionEND.exec</code> to remove the system privilege for the company address entry.

3.2.4.1.1.3 Modify

When modifying properties, Identity Management executes the defined Modify process to perform the modifications in the connected back end system(s). The Modify process has only one internal execution task, namely the Split on Entry Type task.

📘 Note

In case that you customize a parent process (which includes other processes in it) that calls this *Modify* process, the *Use Context Variables* checkbox on the *General* tab of the process must be selected.



3.2.4.1.1.3.1 Split on Entry Type (Modify)

This task (internal execution task), is a switch task that uses the Java library method `com.sap.idm.pf.modify.ExtSplitObjTypeModify.exec` to determine which operation should be executed for each of the entry types `MX_PERSON`, `MX_GROUP` and `MX_COMPANY_ADDRESS`.

- For `MX_PERSON`, the task inspects the modified data and defines a set of context variables based on the findings as the result. The *Execute plugins* task performs a set of checks on these context variables and determines the operations:
 - If context variable `MARK_EXEC_PASSWORD_SET` is set, then proceed with task *8. Exec Plugin - Set Password*. If not set, do nothing.
 - If context variable `MARK_EXEC_ENABLED` is set, then proceed with task *6. Exec Plugin - Enable User*. If not set, do nothing.
 - If context variable `MARK_EXEC_DISABLED` is set, proceed with task *7. Exec Plugin - Disable User*. If not set, do nothing.
 - If context variable `MARK_EXEC_MODIFY_ATTR` is set, proceed with task *2. Exec Plugin - Modify User - no_waitafter*. If not set, do nothing.

The context variables are set as follows:

Context Variable Name	Description
<code>MARK_EXEC_PASSWORD_SET</code>	Set this context variable if password change is detected, i.e. the data contains changes for attributes <code>MX_PASSWORD</code> , <code>MX_ENCRYPTED_PASSWORD</code> or <code>MX_PASSWORD_DISABLED</code> .
<code>MARK_EXEC_ENABLED</code>	Set this context variable if the modifying operation for attribute <code>MX_DISABLED</code> is <i>DELETE</i> .
<code>MARK_EXEC_DISABLED</code>	Set this context variable if the modifying operation for attribute <code>MX_DISABLED</code> is <i>ADD</i> .
<code>MARK_EXEC_MODIFY_ATTR</code>	Set this context variable if change is detected for any other attributes apart from the ones mentioned above (<code>MX_PASSWORD</code> , <code>MX_ENCRYPTED_PASSWORD</code> , <code>MX_PASSWORD_DISABLED</code> and <code>MX_DISABLED</code>) and the following attributes: <code>SAP_CHANGENUMBER</code> - Any attributes starting with <code>MXREF_</code> (except for attribute <code>MXREF_MX_COMPANY_ADDRESS</code>) - Any attributes starting with <code>MX_NOTIFICATION</code> - Any attributes starting with <code>MX_TRIGGER</code> - Any attributes starting with <code>ACCOUNT</code>

Note

When running the checks, the *Execute plugins* task uses SQL queries in the following format:

```
select COUNT(VarName) from mxpv_audit_variables where AuditID=%AUDITID%  
and VarValue='%MSKEY%' and VarName='<name of the context variable>%MSKEY%'
```

For example:

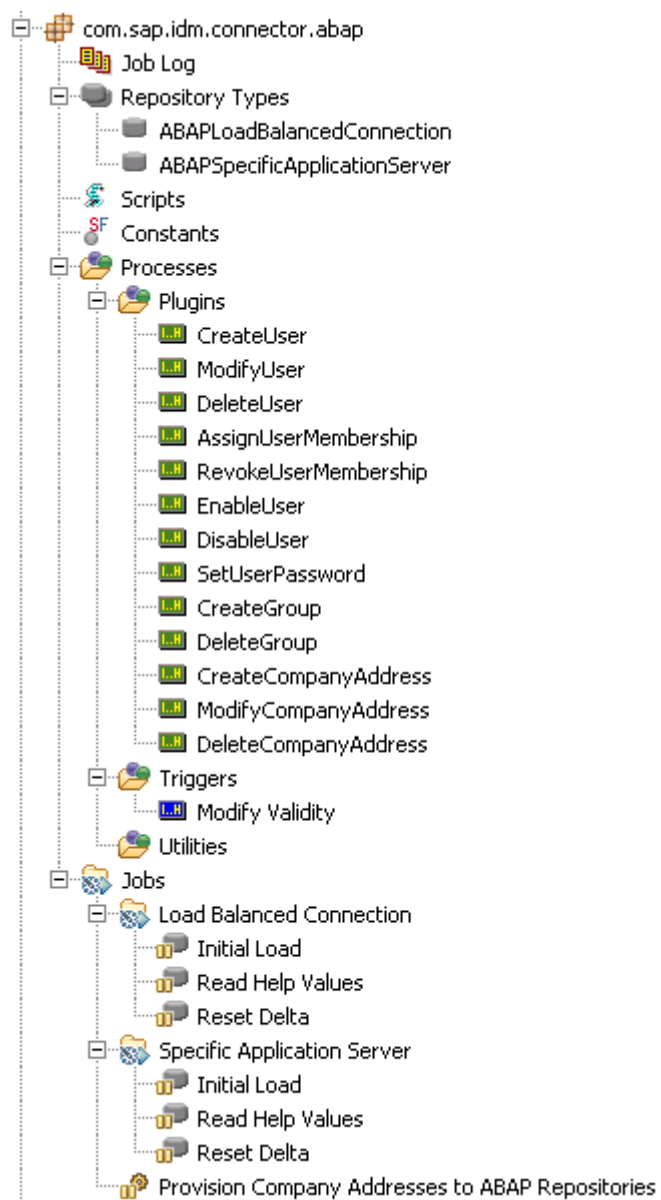
```
select COUNT(VarName) from mxpv_audit_variables where AuditID=%AUDITID%  
and VarValue='%MSKEY%' and VarName='MARK_EXEC_PASSWORD_SET%MSKEY%'
```

- For `MX_GROUP`, the task makes sure that the group privileges are updated:
 - The group privilege is assigned if a new group is assigned to an entry (for example a user).
 - The group privilege for groups that have been deleted is removed.
 - Group references are automatically reassigned to the entry if the group reference has been deleted but the corresponding group privilege was inherited from a business role.
- For `MX_COMPANY_ADDRESS`, proceed with task [13. Exec Plugin - Update Company Address](#).

3.2.4.2 ABAP Connector Package

The ABAP connector package [com.sap.idm.connector.abap](#) contains the SAP Provisioning Framework connector for AS ABAP systems, including a set of connector specific repository types, processes, jobs and scripts.

The structure of this package is as follows:



Repository Types

The ABAP connector package contains the following repository types:

- [*ABAPSpecificApplicationServer*](#)
- [*ABAPLoadBalancedConnection*](#)

The repository type defines common parameters for all repositories based on this type.

Plug-in Processes

The core part of provisioning framework contains the logic to decide which plug-in processes are called, while the connectors contain the plug-in processes that are referenced from the repositories. Each connector has its own set of plug-in processes. The following plug-in processes are used by the ABAP connector:

Plug-in Process	Description
<i>AssignUserMembership</i>	This process is executed when an authorization privilege (without MX_IS_ACCOUNT), is assigned to a user. The user is assigned the corresponding authorization in the repository.
<i>CreateCompanyAddress</i>	This process is triggered upon creation of an entry of type MX_COMPANY_ADDRESS.
<i>CreateGroup</i>	This process is triggered when a group is assigned an account privilege. It creates the group in the repository.
<i>CreateUser</i>	This process is triggered when a user is assigned an account privilege (with the attribute MX_IS_ACCOUNT). The process creates the user account with the mandatory attributes in the repository.
<i>DeleteCompanyAddress</i>	This process is triggered when an MX_COMPANY_ADDRESS entry is deleted.
<i>DeleteGroup</i>	This process is triggered when the account privilege is removed from a group. The group is removed from the repository.
<i>DeleteUser</i>	This process is triggered when the last account privilege is removed from a user. The user may have the account privilege in multiple ways, either via roles or directly assigned. Deprovisioning will only start once the last one has been removed. The account privilege can be removed in two ways. Either directly by removing the privilege from the user, or indirectly when setting attribute MX_INACTIVE on the user. This removes all assignments, including the account privilege. The process inactivates the account in the repository. How this is done depends on the system.

Plug-in Process	Description
<i>DisableUser</i>	This process is triggered when attribute MX_DISABLED is set on the user. This will disable the user's account in the repository. Attribute MX_DISABLED is a global setting, affecting all systems. How a user is actually disabled is system-specific. Some systems have a disable flag, while others change the password for example in order to prevent the user from logging in.
<i>EnableUser</i>	This process is called when attribute MX_DISABLED is removed from the user. This will enable the user's account in the repository. Attribute MX_DISABLED is a global setting, affecting all systems.
<i>ModifyCompanyAddress</i>	The process is triggered by changes for MX_COMPANY_ADDRESS entry.
<i>ModifyUser</i>	This process is triggered when an attribute on a user with the system privilege for the repository is modified. The process updates the repository with all changed attributes.
<i>RevokeUserMembership</i>	This process is triggered when deassigning an authorization privilege from a user. The corresponding authorization in the repository is removed from the user.
<i>SetUserPassword</i>	This process is triggered when attribute MX_ENCRYPTED_PASSWORD is modified and updates the password in the repository. Attribute MX_ENCRYPTED_PASSWORD attribute is a global setting, affecting all systems.

Jobs

The ABAP connector package provides a set of templates that you can use for setting up jobs. The following jobs are supported:

Job	Description
<i>Initial Load</i>	Run this job in order to retrieve the identity information from the connected system and store it in the identity store in the Identity Management database.
<i>Read Help Values</i>	Run this job prior to the initial load in order to read the value help from the ABAP system. The value help content is then written to the corresponding database table specified in the schema.

Job	Description
Reset Delta	Run this job in order to reset the delta information.

Along with the different processes, this package contains several scripts that are used in various cases like converting values, getting roles, getting repository name and others.

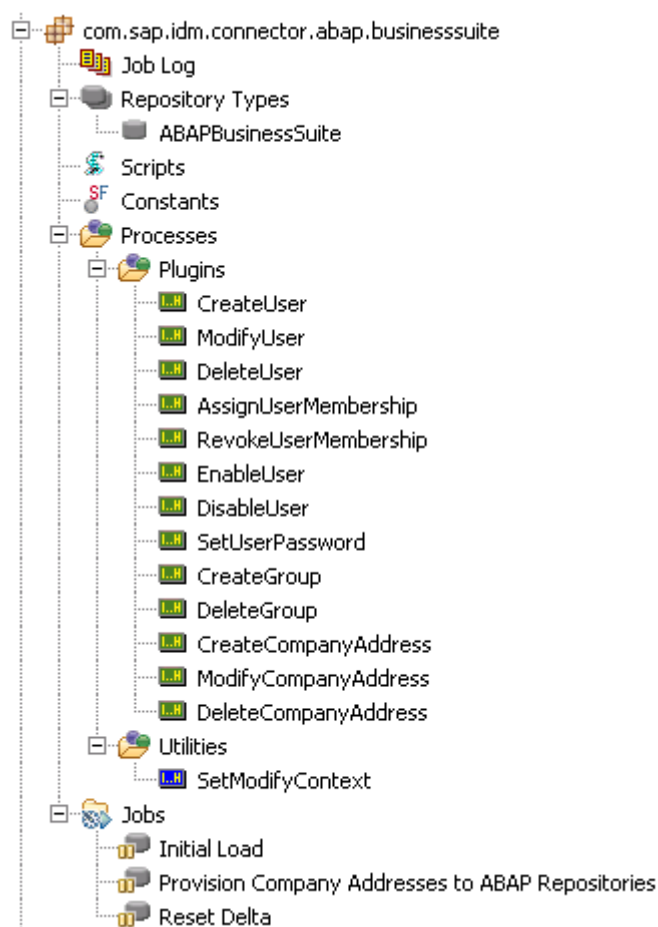
Related Information

[Setting Up an AS ABAP System \[page 623\]](#)

3.2.4.3 ABAP BusinessSuite Connector Package

The ABAP BusinessSuite connector package [com.sap.idm.connector.abap.businesssuite](#) contains the SAP Provisioning Framework connector for SAP Business Suite systems, including a connector specific repository type, processes, jobs and scripts.

The structure of this package is as follows:



Repository Type

The ABAP BusinessSuite connector package contains the following repository type: [ABAPBusinessSuite](#). It defines common parameters for all repositories based on this repository type.

Plug-in Processes

The core part of provisioning framework contains the logic to decide which plug-in process are called, while the connectors contain the plug-in processes that are referenced from the repositories. Each connector has its own set of plug-in processes. The following plug-in processes are used by the ABAP BusinessSuite connector:

Plug-in Process	Description
CreateUser	This process is triggered when a user is assigned an account privilege (with the attribute MX_IS_ACCOUNT). The process creates the user account with the mandatory attributes in the repository.
ModifyUser	This process is triggered when an attribute on a user with the system privilege for the repository is modified. The process updates the repository with all changed attributes. The Business Suite connector uses <code>ToSAPIdentity</code> pass to update the user in the respective Business Suite application, based on the BAdI filters (the function set privilege <code>PRIV:FUNCTION_SET:<Repository>:<BAdI_Filter_Value></code> assigned to the user). Note By default, the BAPI requires that the user exists in the Business Suite application. If the user doesn't exist, the BAPI creates it automatically, even though this is a modify process. For more information, see <i>Process Handling of User Management Tasks</i> in the Extending the SAP Provisioning Framework Developer Guide. If explicitly required, you can change the default behavior, so that the modify process no longer creates a user in the Business Suite application, if the user doesn't exist. See 2950644 .

Plug-in Process	Description
<i>DeleteUser</i>	This process is triggered when the last account privilege is removed from a user. The user may have the account privilege in multiple ways, either via roles or directly assigned. The deprovisioning will only start when the last one is removed. The account privilege can be removed in two ways. Either directly by removing the privilege from the user, or indirectly when setting attribute MX_INACTIVE on the user. This removes all assignments, including the account privilege. The process inactivates the account in the repository. How this is done depends on the system.
<i>AssignUserMembership</i>	This process is executed when an authorization privilege (without MX_IS_ACCOUNT), is assigned to a user. The user is assigned the corresponding authorization in the repository.
<i>RevokeUserMembership</i>	This process is triggered when deassigning an authorization privilege from a user. The corresponding authorization in the repository is removed from the user.
<i>EnableUser</i>	This process is called when attribute MX_DISABLED is removed from the user. This will enable the user's account in the repository. The MX_DISABLED attribute is a global setting, affecting all systems.
<i>DisableUser</i>	This process is triggered when attribute MX_DISABLED is set on the user. This will disable the user's account in the repository. The MX_DISABLED attribute is a global setting, affecting all systems. How a user is actually disabled is system-specific. Some systems may have a disable flag, while others may for example change the password to prevent the user from logging in.
<i>SetUserPassword</i>	This process is triggered when attribute MX_ENCRYPTED_PASSWORD is modified and updates the password in the repository. The MX_ENCRYPTED_PASSWORD attribute is a global setting, involving all systems.
<i>CreateGroup</i>	This process is triggered when a group is assigned an account privilege and creates the group in the repository.
<i>DeleteGroup</i>	This process is triggered when the account privilege is removed from a group. The group is removed from the repository.

Utilities

SetModifyContext process

Jobs

The ABAP BusinessSuite connector package provides a set of templates that you can use for setting up jobs. The following jobs are supported:

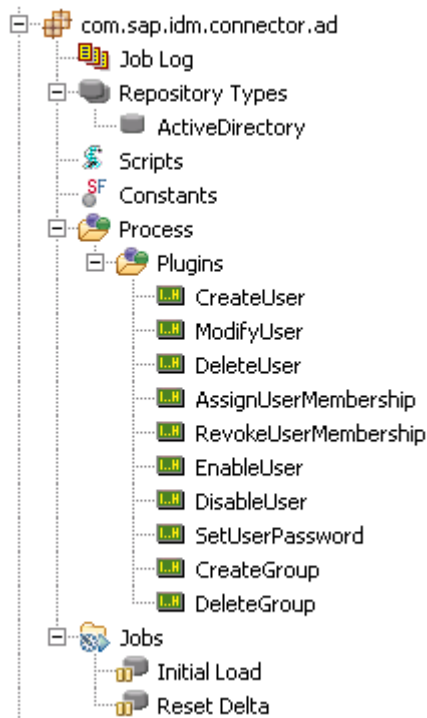
Job	Description
<i>Initial Load</i>	Run this job to retrieve the identity information from the connected system and store it in the identity store in the Identity Management database.
<i>Provision Company Addresses to ABAP Repositories</i>	Run this job to provision the company addresses read during the initial load from the ABAP source system to other connected ABAP systems.
<i>Reset Delta</i>	Run this job to reset the delta information.

Along with the different processes, this package contains several scripts that are used in various cases like verifying help values and others.

3.2.4.4 AD Connector Package

The AD connector package [com.sap.idm.connector.ad](#) contains the SAP Provisioning Framework connector for Microsoft Active Directory Server, including a connector specific repository type, processes, jobs and scripts.

The structure of this package is as follows:



Repository Type

The AD connector package contains the following repository type: [ActiveDirectory](#). It defines common parameters for all repositories based on this repository type.

Plug-in Processes

The core part of the provisioning framework contains the logic to decide which plug-in processes are called, while the connectors contain the plug-in processes that are referenced from the repositories. Each connector has its own set of plug-in processes. The following plug-in processes are used by the AD connector:

Plug-in Process	Description
AssignUserMembership	This process is executed when an authorization privilege (without MX_IS_ACCOUNT), is assigned to a user. The user is assigned the corresponding authorization in the repository.
CreateGroup	This process is triggered when a group is assigned an account privilege and creates the group in the repository.
CreateUser	This process is triggered when a user is assigned an account privilege (with the attribute MX_IS_ACCOUNT). The process creates the user account with the mandatory attributes in the repository.
DeleteGroup	This process is triggered when the account privilege is removed from a group. The group is removed from the repository.

Plug-in Process	Description
<i>DeleteUser</i>	This process is triggered when the last account privilege is removed from a user. The user may have the account privilege in various ways, either via roles or directly assigned. Deprovisioning will not start until the last one is removed. The account privilege can be removed in two ways. Either directly by removing the privilege from the user, or indirectly when setting the MX_INACTIVE attribute on the user. This removes all assignments, including the account privilege. The process inactivates the account in the repository. How this is done depends on the system.
<i>DisableUser</i>	This process is triggered when attribute MX_DISABLED is set on the user. This will disable the user's account in the repository. Attribute MX_DISABLED is a global setting that affects all systems. How a user is actually disabled depends on the system. Some systems might have a disable flag, while others might change the password to prevent the user from logging in for example.
<i>EnableUser</i>	This process is called when attribute MX_DISABLED is removed from the user. This will enable the user's account in the repository. Attribute MX_DISABLED is a global setting that affects all systems.
<i>ModifyUser</i>	This process is triggered when an attribute on a user with the system privilege for the repository is modified. The process updates the repository with all changed attributes.
<i>RevokeUserMembership</i>	This process is triggered when deassigning an authorization privilege from a user. The corresponding authorization in the repository is removed from the user.
<i>SetUserPassword</i>	This process is triggered when attribute MX_ENCRYPTED_PASSWORD is modified and updates the password in the repository. Attribute MX_ENCRYPTED_PASSWORD attribute is a global setting that affects all systems.

Password Maintenance

If there is a password set for the user, it will be provisioned to the Active Directory during [*CreateUser*](#) process and the user will be activated as a Normal User (useraccountcontrol = 512). The same is also valid for [*EnableUser*](#) process - the user will be activated as Normal User.

If a password is not set for the user during [*CreateUser*](#) or [*EnableUser*](#) processes, the user will be set as Active User With No Password Required (useraccountcontrol = 544) in the Active Directory.

⚠ Caution

That means the password for the user in the Active Directory will be blank.

In this case, the following warning message is added to the job log: "The user has been activated with 'no password required' flag set!". This scenario is kept for backward compatibility and covering some customer cases in which setting password in the Active Directory might not be required.

Jobs

The AD connector package provides a set of templates that you can use for setting up jobs. The following jobs are supported:

Job	Description
<i>Initial Load</i>	Run this job to retrieve the identity information from the connected system and store it in the identity store in the Identity Management database.
<i>Reset Delta</i>	Run this job to reset the delta information.

Along with the different processes, this package contains several scripts used in various cases like generating display names, getting privilege names and others.

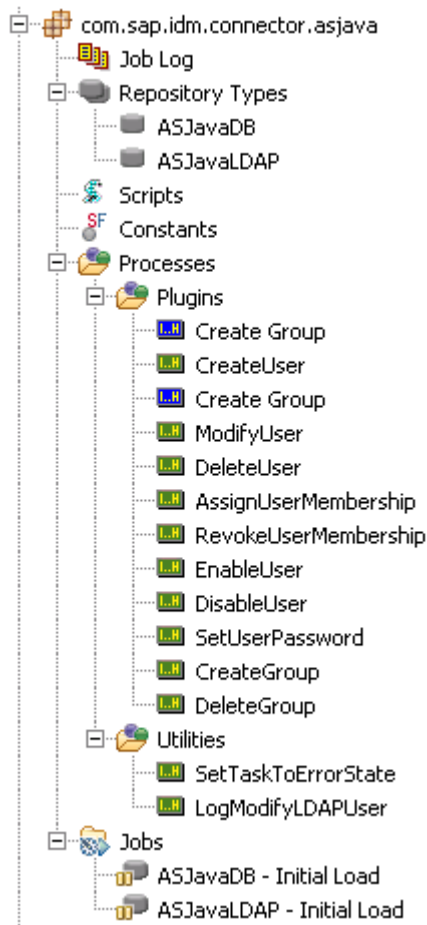
Related Information

[Setting Up a Microsoft Active Directory and Microsoft Exchange \[page 679\]](#)

3.2.4.5 AS Java Connector Package

The AS Java connector package [com.sap.idm.connector.asjava](#) contains the SAP Provisioning Framework connector for SAP NetWeaver AS for Java systems, including a set of connector specific repository types, processes, jobs and scripts.

The structure of this package is as follows:



Repository Types

The AS Java connector package contains the following repository types:

- [ASJavaDB](#)
- [ASJavaLDAP](#)

The repository type defines common parameters for all repositories based on this type.

[Plug-in Processes](#)

The core part of provisioning framework contains the logic to decide which plug-in process are called, while the connectors contain the plug-in processes that are referenced from the repositories. Each connector has its own set of plug-in processes. The following plug-in processes are used by the AS Java connector:

Plug-in Process	Description
<i>AssignUserMembership</i>	This process is executed when an authorization privilege (without MX_IS_ACCOUNT), is assigned to a user. The user is assigned the corresponding authorization in the repository.
<i>CreateGroup</i>	This process is triggered when a group is assigned an account privilege and creates the group in the repository.
<i>CreateUser</i>	This process is triggered when a user is assigned an account privilege (with the attribute MX_IS_ACCOUNT). The process creates the user account with the mandatory attributes in the repository.
<i>DeleteGroup</i>	This process is triggered when the account privilege is removed from a group. The group is removed from the repository.
<i>DeleteUser</i>	This process is triggered when the last account privilege is removed from a user. The user may have the account privilege in multiple ways, either via roles or directly assigned. The deprovisioning will only start when the last one is removed. The account privilege can be removed in two ways. Either directly by removing the privilege from the user, or indirectly when setting attribute MX_INACTIVE on the user. This removes all assignments, including the account privilege. The process inactivates the account in the repository. How this is done depends on the system.
<i>DisableUser</i>	This process is triggered when attribute MX_DISABLED is set on the user. This will disable the user's account in the repository. The MX_DISABLED attribute is a global setting, affecting all systems. How a user is actually disabled is system-specific. Some systems may have a disable flag, while others may for example change the password to prevent the user from logging in.
<i>EnableUser</i>	This process is called when attribute MX_DISABLED is removed from the user. This will enable the user's account in the repository. The MX_DISABLED attribute is a global setting, affecting all systems.
<i>ModifyUser</i>	This process is triggered when an attribute on a user with the system privilege for the repository is modified. The process updates the repository with all changed attributes.

Plug-in Process	Description
<i>RevokeUserMembership</i>	This process is triggered when deassigning an authorization privilege from a user. The corresponding authorization in the repository is removed from the user.
<i>SetUserPassword</i>	This process is triggered when attribute MX_ENCRYPTED_PASSWORD is modified and updates the password in the repository. The MX_ENCRYPTED_PASSWORD attribute is a global setting, involving all systems.

Jobs

The AS Java connector package provides a set of templates that you can use for setting up jobs. The following jobs are supported:

Job	Description
<i>ASJavaDB - Initial Load</i>	Run this job to retrieve the identity information from the connected system and store it in the identity store in the Identity Management database.
<i>ASJavaLDAP - Initial Load</i>	Run this job to retrieve the identity information from the connected system and store it in the identity store in the Identity Management database.

Along with the different processes, this package contains several scripts used in various cases like generating SPML IDs, getting group account values and others.

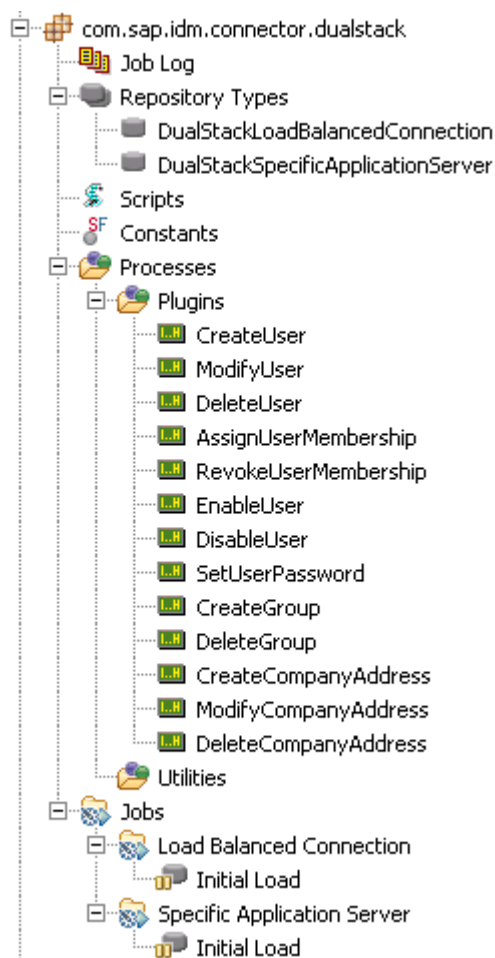
Related Information

[Setting Up an AS Java System \[page 628\]](#)

3.2.4.6 Dual Stack Connector Package

The dual stack connector package [com.sap.idm.connector.dualstack](#) contains the SAP Provisioning Framework connector for dual-stack systems (AS ABAP/AS Java), including a set of connector specific repository types and processes.

The structure of this package is as follows:



Repository Types

The dual-stack connector package contains the following repository types:

- [*DualStackSpecificApplicationServer*](#)
- [*DualStackLoadBalancedConnection*](#)

The repository type defines common parameters for all repositories based on this type.

[*Plug-in Processes*](#)

The core part of provisioning framework contains the logic to decide which plug-in process are called, while the connectors contain the plug-in processes that are referenced from the repositories. Each connector has its own set of plug-in processes. Each connector plug-in process is referenced by the repository type. The following plug-in processes are used by the dual-stack connector:

Plug-in Process	Description
<i>AssignUserMembership</i>	This process is executed when an authorization privilege (without MX_IS_ACCOUNT), is assigned to a user. The user is assigned the corresponding authorization in the repository.
<i>CreateCompanyAddress</i>	This process is triggered upon creation of an entry of type MX_COMPANY_ADDRESS.
<i>CreateGroup</i>	This process is triggered when a group is assigned an account privilege and creates the group in the repository.
<i>CreateUser</i>	This process is triggered when a user is assigned an account privilege (with the attribute MX_IS_ACCOUNT). The process creates the user account with the mandatory attributes in the repository.
<i>DeleteCompanyAddress</i>	This process is triggered when MX_COMPANY_ADDRESS entry is deleted.
<i>DeleteGroup</i>	This process is triggered when the account privilege is removed from a group. The group is removed from the repository.
<i>DeleteUser</i>	This process is triggered when the last account privilege is removed from a user. The user may have the account privilege in multiple ways, either via roles or directly assigned. The deprovisioning will only start when the last one is removed. The account privilege can be removed in two ways. Either directly by removing the privilege from the user, or indirectly when setting attribute MX_INACTIVE on the user. This removes all assignments, including the account privilege. The process inactivates the account in the repository. How this is done depends on the system.
<i>DisableUser</i>	This process is triggered when attribute MX_DISABLED is set on the user. This will disable the user's account in the repository. The MX_DISABLED attribute is a global setting, affecting all systems. How a user is actually disabled is system-specific. Some systems may have a disable flag, while others may for example change the password to prevent the user from logging in.
<i>EnableUser</i>	This process is called when attribute MX_DISABLED is removed from the user. This will enable the user's account in the repository. The MX_DISABLED attribute is a global setting, affecting all systems.

Plug-in Process	Description
<i>ModifyCompanyAddress</i>	The process is triggered by changes for MX_COMPANY_ADDRESS entry.
<i>ModifyUser</i>	This process is triggered when an attribute on a user with the system privilege for the repository is modified. The process updates the repository with all changed attributes.
<i>RevokeUserMembership</i>	This process is triggered when deassigning an authorization privilege from a user. The corresponding authorization in the repository is removed from the user.
<i>SetUserPassword</i>	This process is triggered when attribute MX_ENCRYPTED_PASSWORD is modified and updates the password in the repository. The MX_ENCRYPTED_PASSWORD attribute is a global setting, involving all systems.

Jobs

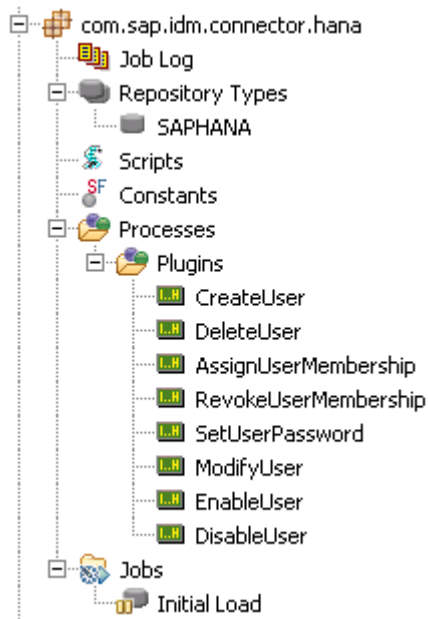
The dual stack connector package provides a set of templates that you can use for setting up jobs. The following jobs are supported:

Job	Description
<i>Initial Load</i>	Run this job in order to retrieve the identity information from the connected system and store it in the identity store in the Identity Management database.

3.2.4.7 HANA Connector Package

The HANA connector package [*com.sap.idm.connector.hana*](#) contains the SAP Provisioning Framework connector for SAP HANA, including a connector-specific repository type, processes, jobs and scripts.

The structure of this package is as follows:



Repository Type

The HANA connector package contains the following repository type: [SAPHANA](#). It defines common parameters for all repositories based on this repository type.

Plug-in Processes

The core part of provisioning framework contains the logic to decide which plug-in processes are called, while the connectors contain the plug-in processes that are referenced from the repositories. Each connector has its own set of plug-in processes. The following plug-in processes are used by the HANA connector:

Plug-in Process	Description
AssignUserMembership	This process is executed when an authorization privilege (without MX_IS_ACCOUNT), is assigned to a user. The user is assigned the corresponding authorization in the repository.
CreateUser	This process is triggered when a user is assigned an account privilege (with the attribute MX_IS_ACCOUNT). The process creates the user account with the mandatory attributes in the repository.

Plug-in Process	Description
<i>DeleteUser</i>	This process is triggered when the last account privilege is removed from a user. The user can have the account privilege in various ways, either via roles or directly assigned. Deprovisioning will not start until the last one is removed. The account privilege can be removed in two ways. Either directly by removing the privilege from the user, or indirectly when setting the MX_INACTIVE attribute on the user. This removes all assignments, including the account privilege. The process inactivates the account in the repository. How this is done depends on the system.
<i>DisableUser</i>	This process is triggered when attribute MX_DISABLED is set on the user. This will disable the user's account in the repository. Attribute MX_DISABLED is a global setting that affects all systems. How a user is actually disabled depends on the system. Some systems might have a disable flag, while others might change the password to prevent the user from logging in for example.
<i>EnableUser</i>	This process is called when attribute MX_DISABLED is removed from the user. This will enable the user's account in the repository. Attribute MX_DISABLED is a global setting that affects all systems.
<i>ModifyUser</i>	This process is triggered when an attribute on a user with the system privilege for the repository is modified. The process updates the repository with all changed attributes. Note Although you can comment out (disable) attributes in the Modify SAP HANA User pass, the default value of the attributes will be provisioned due to the HANA specific behavior.
<i>RevokeUserMembership</i>	This process is triggered when deassigning an authorization privilege from a user. The corresponding authorization in the repository is removed from the user.
<i>SetUserPassword</i>	This process is triggered when attribute MX_ENCRYPTED_PASSWORD is modified and updates the password in the repository. Attribute MX_ENCRYPTED_PASSWORD attribute is a global setting that affects all systems.

Jobs

The HANA connector package provides a template that you can use for setting up jobs. The following job is available:

- Initial Load - Run this job to retrieve the identity information from the connected system and store it in the identity store in the Identity Management database.
When you run the HANA initial load job for the first time, you get a warning in the following section of the job log: [Pass: Read SAP HANA features](#). According to the warning the following variables were not defined in the repository: `<VALID_FROM>`, `<USER_PARAMETERS>`, `<X509_USER_MAPPINGS>`, `<SAML_USER_MAPPINGS>`, `<IS_SAP_LOGON_TICKET_ENABLED>`, `<IS_SAP_ASSERTION_TICKET_ENABLED>`. Regardless of the warning, the variables are created and you can continue with the next step. This warning appears only with the first execution of the initial load.
Repository variables are available only from scripts. To modify or create a repository variable, use the `uSetRepositoryVar` internal function. To get a repository variable, use the `uGetRepositoryVar` internal function.

Along with the different processes, the HANA connector package contains several scripts used in various cases like generating display names, getting parameter account values and others.

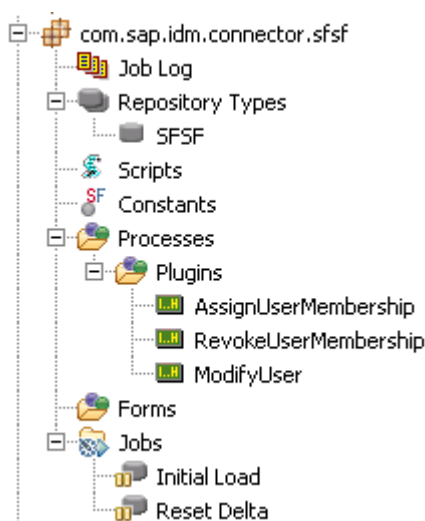
Related Information

[Setting Up an SAP HANA System \[page 632\]](#)

3.2.4.8 Success Factors Connector Package

The Success Factors connector package [com.sap.idm.connector.sfsf](#) contains the SAP Provisioning Framework connector for Success Factors, including a connector specific repository type, processes, jobs and scripts.

The structure of this package is as follows:



Repository Type

The Success Factors connector package contains the following repository type: [SFSF](#). It defines common parameters for all repositories based on this repository type.

Plug-in Processes

The core part of the provisioning framework contains the logic used to decide which plug-in processes are called, while the connectors contain the plug-in processes that are referenced from the repositories. Each connector has its own set of plug-in processes. The following plug-in processes are used by the Success Factors connector:

Plug-in Process	Description
AssignUserMembership	This process is executed when an authorization privilege (without MX_IS_ACCOUNT), is assigned to a user. The user is assigned the corresponding authorization in the repository.
ModifyUser	This process is triggered when an attribute on a user with the system privilege for the repository is modified. The process updates the repository with all changed attributes. Note You can modify the username and the email address of the user, that is, the MX_HCM_SYSUNAME and the MX_MAIL_PRIMARY attributes. Changes to other user attributes in SAP Identity Management are not provisioned to SuccessFactors.
RevokeUserMembership	This process is triggered when deassigning an authorization privilege from a user. The corresponding authorization in the repository is removed from the user.

Jobs

The Success Factors connector package provides a set of templates that you can use for setting up jobs. The following jobs are supported:

Job	Description
Initial Load	Run this job to retrieve the identity information from the connected system and to store it in the identity store in the identity management database. This job can load inactive users that are in the following status: • Retired • Terminated • Discarded • Furlough Inactive users are stored in the identity store as disabled. If you want to configure the Read SFSF Employees (SOAP) pass in this Initial Load job, see <i>Parameters for FromSFSFSoap Pass</i>.
Reset Delta	Run this job to reset the delta information.

Along with the various processes, this package contains several scripts used in various cases like generating target group IDs and acquiring country codes.

Related Information

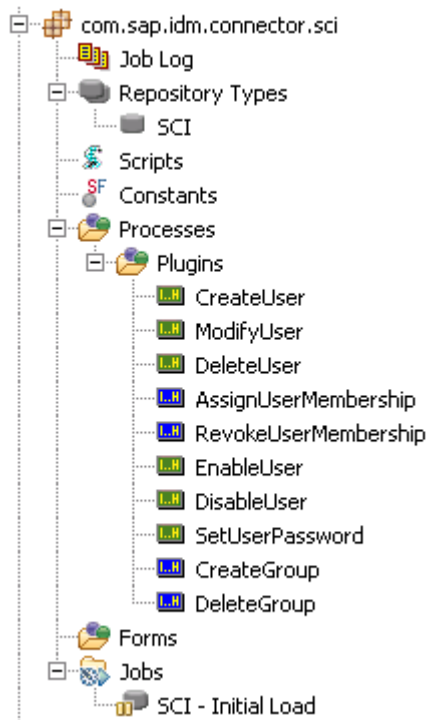
[Setting Up a SuccessFactors System \[page 640\]](#)

[Parameters for FromSFSFSoap Pass \[page 1039\]](#)

3.2.4.9 Identity Authentication Connector Package

The Identity Authentication connector package [com.sap.idm.connector.sci](#) contains the SAP Provisioning Framework connector for the SAP Cloud Identity Services - Identity Authentication, formerly known as SAP Cloud Identity (SCI). The package includes a connector specific repository type, processes, jobs and scripts.

The structure of this package is as follows:



Note

Currently, only [Identity Authentication SCIM REST API](#) is supported. As stated in the documentation, this API is deprecated. If you would like to use the [Identity Directory SCIM REST API](#), please use the [hybrid integration with the Cloud Systems via the IPS Proxy](#). You can set which API version you would like to use with a property on the IPS Proxy system. For more information, see [Identity Authentication](#).

Repository Type

The Identity Authentication connector package contains the following repository type: [SCI](#). It defines common parameters for all repositories based on this repository type.

Plug-in Processes

The core part of provisioning framework contains the logic to decide which plug-in processes are called, while the connectors contain the plug-in processes that are referenced from the repositories. Each connector has its own set of plug-in processes. The following plug-in processes are used by the Identity Authentication connector:

Plug-in Process	Description
CreateUser	This process is triggered when a user is assigned an account privilege (with the attribute MX_IS_ACCOUNT). The process creates the user account with the mandatory attributes in the repository. Note If you create a user with MX_DEPARTMENT attribute defined for this user, make sure the department exists. That is, the attribute must have a valid value. Otherwise, the CreateUser process will fail. Note If you create a user without setting a productive password, make sure you provide a valid email address for this user. In this case, the user will receive an account activation email.
DeleteUser	This process is triggered when the last account privilege is removed from a user. The user may have the account privilege in various ways, either via roles or directly assigned. Deprovisioning will not start until the last one is removed. The account privilege can be removed in two ways. Either directly by removing the privilege from the user, or indirectly when setting the MX_INACTIVE attribute on the user. This removes all assignments, including the account privilege. The process inactivates the account in the repository. How this is done depends on the system.
DisableUser	This process is triggered when attribute MX_DISABLED is set on the user. This will disable the user's account in the repository. Attribute MX_DISABLED is a global setting that affects all systems. How a user is actually disabled depends on the system. Some systems might have a disable flag, while others might change the password to prevent the user from logging in for example.
EnableUser	This process is called when attribute MX_DISABLED is removed from the user. This will enable the user's account in the repository. Attribute MX_DISABLED is a global setting that affects all systems.
ModifyUser	This process is triggered when an attribute on a user with the system privilege for the repository is modified. The process updates the repository with all changed attributes.

Plug-in Process	Description
<i>SetUserPassword</i>	This process is triggered when attribute MX_ENCRYPTED_PASSWORD is modified and updates the password in the repository. The MX_ENCRYPTED_PASSWORD attribute is a global setting, involving all systems.

Jobs

The Identity Authentication connector package provides a set of templates that you can use for setting up jobs. The following job is supported:

Job	Description
<i>Initial Load</i>	Run this job to retrieve the identity information from the connected system and to store it in the identity store in the identity management database.

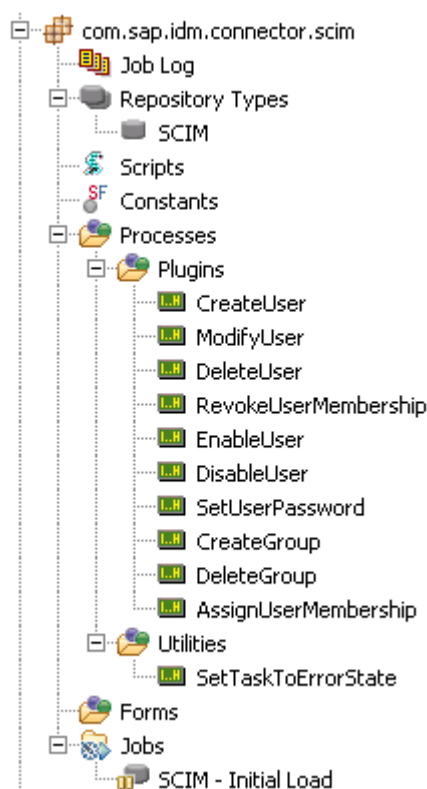
Related Information

[Setting up an Identity Authentication System \[page 645\]](#)

3.2.4.10 SCIM Connector Package

The SCIM connector package [com.sap.idm.connector.scim](#) contains the SAP Provisioning Framework connector for SCIM 2.0 based systems, used for identity provisioning. It includes a specific repository type and a set of processes, jobs and scripts.

The structure of this package is as follows:



Repository Type

The SCIM connector package contains the following repository type: [SCIM](#). It defines common parameters for all repositories based on this repository type.

Plug-in Processes

The core part of provisioning framework contains the logic to decide which plug-in processes are called, while the connectors contain the plug-in processes that are referenced from the repositories. Each connector has its own set of plug-in processes. The following plug-in processes are used by the SCIM connector:

Plug-in Process	Description
CreateUser	This process is triggered when a user is assigned an account privilege (with the attribute MX_IS_ACCOUNT). The process creates the user account with the mandatory attributes in the repository.

Plug-in Process	Description
ModifyUser	This process is triggered when an attribute on a user with the system privilege for the repository is modified. The process updates the repository with all changed attributes.
DeleteUser	This process is triggered when the last account privilege is removed from a user. The user may have the account privilege in various ways, either via roles or directly assigned. Deprovisioning will not start until the last one is removed. The account privilege can be removed in two ways. Either directly by removing the privilege from the user, or indirectly when setting the MX_INACTIVE attribute on the user. This removes all assignments, including the account privilege. The process inactivates the account in the repository. How this is done depends on the system.
RevokeUserMembership	This process is triggered when a group or a group privilege is deassigned from a user or a group. The corresponding authorization in the repository is removed from the user or the group.
EnableUser	This process is called when attribute MX_DISABLED is removed from the user. This will enable the user's account in the repository. Attribute MX_DISABLED is a global setting that affects all systems. The EnableUser process triggers the ModifyUser process.
DisableUser	This process is triggered when attribute MX_DISABLED is set on the user. This will disable the user's account in the repository. Attribute MX_DISABLED is a global setting that affects all systems. How a user is actually disabled depends on the system. Some systems might have a disable flag, while others might change the password to prevent the user from logging in for example. The DisableUser process triggers the ModifyUser process.
SetUserPassword	This process is triggered when attribute MX_ENCRYPTED_PASSWORD is modified and updates the password in the repository. The MX_ENCRYPTED_PASSWORD attribute is a global setting, involving all systems.
CreateGroup	This process is triggered when a group is assigned an account privilege and creates the group in the repository.

Plug-in Process	Description
<i>DeleteGroup</i>	This process is triggered when the account privilege is removed from a group. The group is removed from the repository.
<i>AssignUserMembership</i>	This process is executed when a group or a group privilege is assigned to a user or a group. The user or the group is assigned the corresponding authorization in the repository.

Jobs

The SCIM connector supports the following job:

Job	Description
<i>Initial Load</i>	Run this job to retrieve the identity information from the connected system and to store it in the identity store in the identity management database.

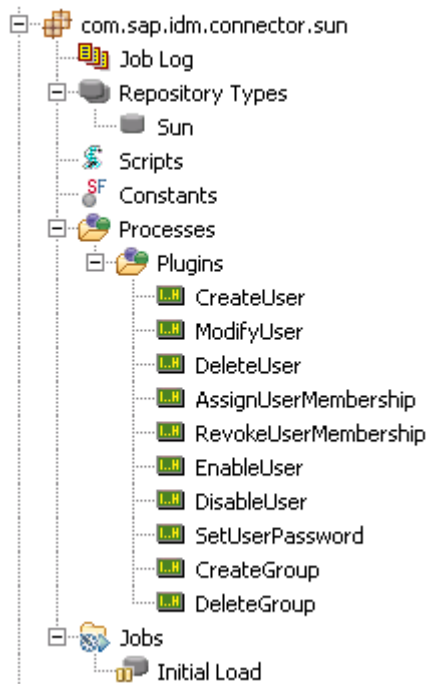
Related Information

[Setting up a SCIM System \[page 646\]](#)

3.2.4.11 Sun Connector Package

The Sun connector package [com.sap.idm.connector.sun](#) contains the SAP Provisioning Framework connector for SUN LDAP Server, including a connector-specific repository type, processes and jobs.

The structure of this package is as follows:



Repository Types

The Sun connector package contains one repository type: [Sun](#). It defines common parameters for all repositories based on this repository type.

Plug-in Processes

The core part of the provisioning framework contains the logic to decide which plug-in processes are called, while the connectors contain the plug-in processes that are referenced from the repositories. Each connector has its own set of plug-in processes. The following plug-in processes are used by the Sun connector:

Plug-in Process	Description
AssignUserMembership	This process is executed when an authorization privilege (without MX_IS_ACCOUNT), is assigned to a user. The user is assigned the corresponding authorization in the repository.
CreateGroup	This process is triggered when a group is assigned an account privilege and creates the group in the repository.

Plug-in Process	Description
<i>CreateUser</i>	This process is triggered when a user is assigned an account privilege (with the attribute MX_IS_ACCOUNT). The process creates the user account with the mandatory attributes in the repository.
<i>DeleteGroup</i>	This process is triggered when the account privilege is removed from a group. The group is removed from the repository.
<i>DeleteUser</i>	This process is triggered when the last account privilege is removed from a user. The user can have the account privilege in various ways, either via roles or directly assigned. The deprovisioning will only start when the last account privilege is removed. The account privilege can be removed in two ways. Either directly by removing the privilege from the user, or indirectly when setting the MX_INACTIVE attribute on the user. This removes all assignments, including the account privilege. The process inactivates the account in the repository. How this is done depends on the system.
<i>DisableUser</i>	This process is triggered when attribute MX_DISABLED is set on the user. This will disable the user's account in the repository. Attribute MX_DISABLED is a global setting that affects all systems. How a user is actually disabled depends on the system. Some systems may a disable flag, while others change the password to prevent the user from logging in for example.
<i>EnableUser</i>	This process is called when attribute MX_DISABLED is removed from the user. This will enable the user's account in the repository. Attribute MX_DISABLED attribute is a global setting that affects all systems.
<i>ModifyUser</i>	This process is triggered when an attribute on a user with the system privilege for the repository is modified. The process updates the repository with all changed attributes.
<i>RevokeUserMembership</i>	This process is triggered when deassigning an authorization privilege from a user. The corresponding authorization in the repository is removed from the user.
<i>SetUserPassword</i>	This process is triggered when attribute MX_ENCRYPTED_PASSWORD is modified, and updates the password in the repository. Attribute MX_ENCRYPTED_PASSWORD is a global setting that affects all systems.

Jobs

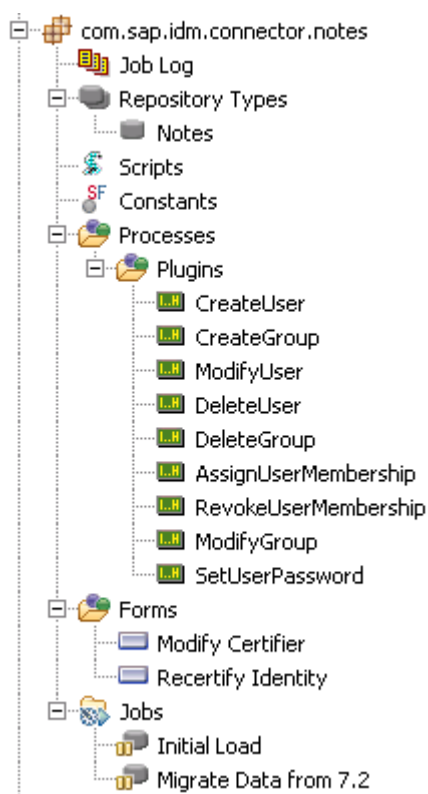
The Sun connector package provides a template that you can use for setting up jobs. The following job is available:

- Initial Load - Run this job to retrieve the identity information from the connected system and store it in the identity store in the Identity Management database.

3.2.4.12 Lotus Notes Connector Package

The Lotus Notes connector package [com.sap.idm.connector.notes](#) contains the SAP Provisioning Framework connector for IBM Lotus Notes systems, including a set of connector specific repository type, processes, forms, jobs and scripts.

The structure of this package is as follows:



Repository Types

The Lotus Notes connector package contains the following repository type: [Notes](#). It defines common parameters for all repositories based on this repository type.

Plug-in Processes

The core part of provisioning framework contains the logic to decide which plug-in processes are called, while the connectors contain the plug-in processes that are referenced from the repositories. Each connector has its own set of plug-in processes. The following plug-in processes are used by the Lotus Notes connector:

Plug-in Processes	Description
<i>AssignUserMembership</i>	This process is executed when an authorization privilege (without MX_IS_ACCOUNT), is assigned to a user. The user is assigned the corresponding authorization in the repository.
<i>CreateGroup</i>	This process is triggered when a group is assigned an account privilege and creates the group in the repository.
<i>CreateUser</i>	This process is triggered when a user is assigned an account privilege (with the attribute MX_IS_ACCOUNT). The process creates the user account with the mandatory attributes in the repository.
<i>DeleteGroup</i>	This process is triggered when the account privilege is removed from a group. The group is removed from the repository.
<i>DeleteUser</i>	This process is triggered when the last account privilege is removed from a user. The user may have the account privilege in multiple ways, either via roles or directly assigned. The deprovisioning will only start when the last one is removed. The account privilege can be removed in two ways. Either directly by removing the privilege from the user, or indirectly when setting attribute MX_INACTIVE on the user. This removes all assignments, including the account privilege. The process inactivates the account in the repository. How this is done depends on the system.
<i>ModifyGroup</i>	This process is triggered when an attribute on a group with the system privilege for the repository is modified. The process updates the repository with all changed attributes.
<i>ModifyUser</i>	This process is triggered when an attribute on a user with the system privilege for the repository is modified. The process updates the repository with all changed attributes.
<i>RevokeUserMembership</i>	This process is triggered when deassigning an authorization privilege from a user. The corresponding authorization in the repository is removed from the user.

Forms

The Lotus Notes connector package is delivered with two forms (UI tasks) for adjusting Lotus Notes certifiers and recertifying Lotus Notes users. You can access them at: `http://<host>:<port>/idm:`

- [Modify Certifier](#) - A form that modifies entries of entry type SAP_NOTES_CERTIFIER.
- [Recertify Identity](#) - A form that recertifies Lotus Notes users and specifies new expiration date for the given users.

Jobs

The following jobs are supported:

Job	Description
Initial Load	Run this job to retrieve the identity information from the connected system and store it in the identity store in the Identity Management database. The job loads users, groups, assignments, certifiers, server and policy information from Lotus Notes. For details about SAP_NOTES_CERTIFIER, SAP_NOTES_SERVER and SAP_NOTES_POLICY entry types and their attributes, see Lotus Notes Schema [page 684].
Migrate Data from 7.2	Run this job to migrate all certifier, policy and server entries from your 7.2 system to your 8.0 system. This job is relevant only for upgrade scenario from version 7.2 to 8.0. The job is available as of SAP Identity Management 8.0 SP02. For more information, see Migrating to Java-Based Lotus Notes Connector [page 693].

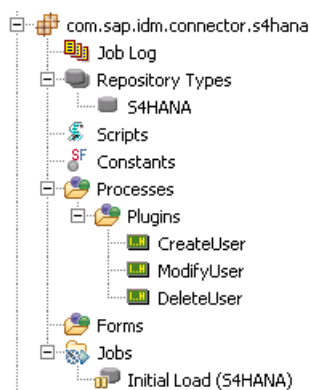
Related Information

[Setting Up a Lotus Notes System \[page 682\]](#)

3.2.4.13 S4HANA Connector Package

The S4HANA connector package [com.sap.idm.connector.s4hana](#) contains the SAP Provisioning Framework connector for SAP S/4HANA on-premise system, including a connector specific repository type, processes, scripts and initial load job.

The structure of this package is as follows:



Repository Types

The S4HANA connector package contains one repository type: [S4HANA](#). It defines common parameters for all repositories based on this repository type.

Plug-in Processes

The core part of the provisioning framework contains the logic to decide which plug-in processes are called, while the connectors contain the plug-in processes that are referenced from the repositories. Each connector has its own set of plug-in processes. The following plug-in processes are used by the S4HANA connector:

Plug-in Process	Description
CreateUser	This process is triggered when a user is assigned an account privilege (with the attribute MX_IS_ACCOUNT). The process creates the user account with the mandatory attributes in the repository.
ModifyUser	This process is triggered when an attribute on a user with the system privilege for the repository is modified. The process updates the repository with all changed attributes.

Plug-in Process	Description
<i>DeleteUser</i>	This process is triggered when the last account privilege is removed from a user. The user can have the account privilege in various ways, either via roles or directly assigned. The deprovisioning will only start when the last account privilege is removed. The account privilege can be removed in two ways. Either directly by removing the privilege from the user, or indirectly when setting the MX_INACTIVE attribute on the user. This removes all assignments, including the account privilege. The process inactivates the account in the repository. How this is done depends on the system.

Jobs

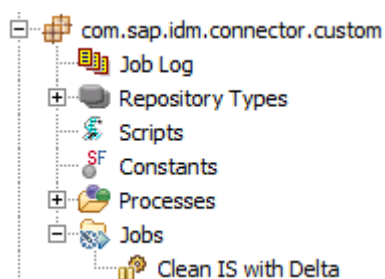
The S4HANA connector package provides a template that you can use for setting up jobs. The following job is available:

Initial Load (S4HANA) - Run this job to retrieve the identity information from the connected system and store it in the identity store in the Identity Management database. For more information, see: [S/4HANA Initial Load Job \[page 654\]](#)

3.2.4.14 Custom Package

The custom package `com.sap.idm.connector.custom` allows you to customize the SAP provisioning framework without the need to change the SAP delivered packages. It contains customizations and extensions made by the customer, default constants, and custom scripts that can be used as templates.

The structure of this package is as follows:



Jobs

The following job is available in the custom package:

- Clean IS with Delta – Run this job to delete all entries, links, values, semaphores, audits and provisioning from the identity store. After running this job, all users will be removed from the identity store, so you have to manually add a new administrator and, if required, other users. For more information, see *Adding a User to the Identity Store*.

⚠ Caution

This job permanently deletes all the listed above records from the identity store. Use it with caution.

The package contains a number of script templates that could be used in various cases, for example to initialize passwords, or to generate user names.

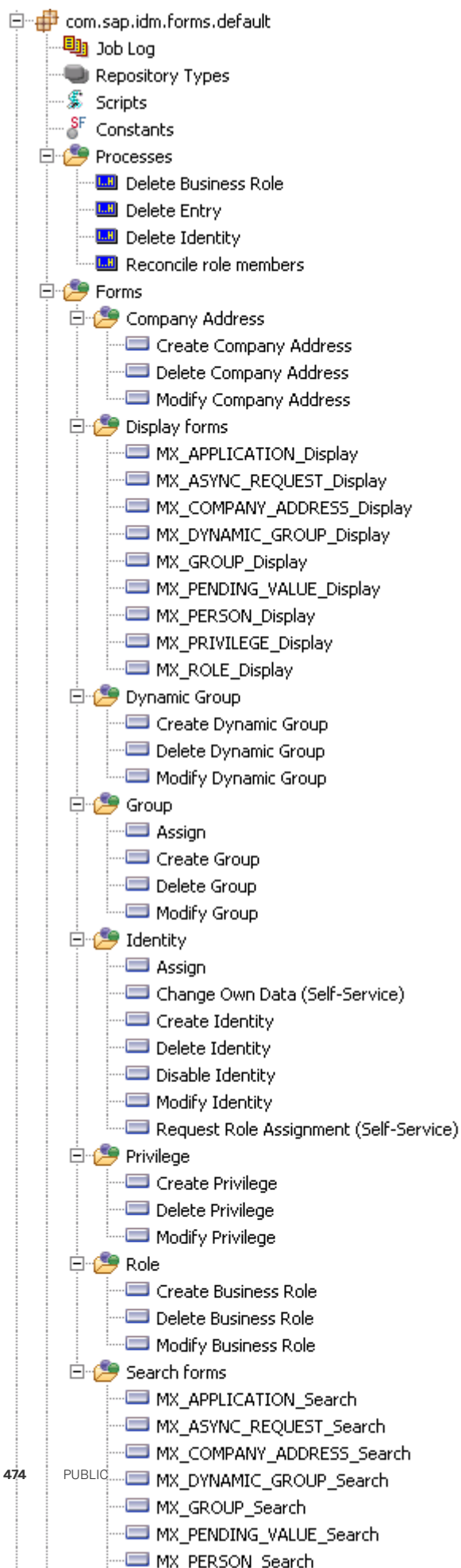
Related Information

[Adding a User to the Identity Store \[page 127\]](#)

3.2.4.15 Forms Package

The forms package `com.sap.idm.forms.default` contains the definition of all user interface tasks for CRUD operations (create, read, update, delete) on different entry types as well as a set of WebDynpro-based sample display and search forms for the entry types, a few processes, package constants and one script.

The structure of this package is as follows:



For more information about the forms and how to maintain forms, see *Maintaining Forms*.

Processes

The forms package contains the following processes:

- Delete Business Role
- Delete Entry
- Delete Identity
- Reconcile role members

Display/Search forms

The forms package contains display/search forms for the following entry types:

- MX_APPLICATION
- MX_ASYNC_REQUEST
- MX_COMPANY_ADDRESS
- MX_DYNAMIC_GROUP
- MX_GROUP
- MX_PENDING_VALUE
- MX_PERSON
- MX_PRIVILEGE
- MX_ROLE

Along with the display/search forms, this package contains forms for creating, deleting, and modifying the following entry types: company address, dynamic group, group, identity, privilege, and role.

The forms package contains a script for the reconciliation of users.

Related Information

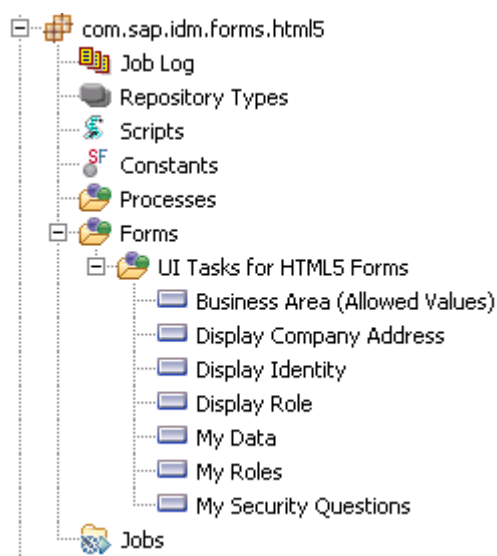
[Forms \[page 71\]](#)

[Maintaining Forms \[page 270\]](#)

3.2.4.16 HTML5 Forms Package

The HTML5 forms package [com.sap.idm.forms.html5](#) contains a set of SAP UI5-based sample forms for displaying different entry types, and forms for managing user attributes.

The structure of this package is as follows:



Display Forms

The HTML5 forms package contains display forms for the following:

- Entry types:
 - MX_COMPANY_ADDRESS
 - MX_PERSON
 - MX_ROLE
- Attribute:
 - MX_BUSINESS_AREA - used to display business area when requesting a role.

Modify Forms

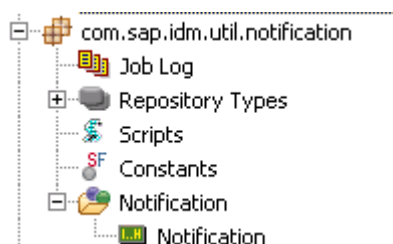
The HTML5 forms package contains forms for modifying the following:

- My Data - Allows modification of personal data.
- My Roles - Allows modification of personal roles.
- My Security Questions - Allows modification of personal security questions.

3.2.4.17 Notification Package

The notification package `com.sap.idm.util.notification` contains the notification process and the notification templates that are used to send notifications from the SAP Provisioning framework, approval and attestation tasks.

The structure of this package is as follows:



As of SAP Note [3108059](#), Java Mail API is used for sending notifications, which gives additional security and encryption capabilities. After applying the Note, the notification package will use the new `uSendMail` function instead of `uSendSMTPMessage`.

Package constants

The following package constants are defined:

Package Constant	Description
DEFAULT_CHAR_ENCODING	Supported encodings US-ASCII,ISO-8859-1,UTF-8,UTF-16BE,UTF-16LE,UTF-16
DEFAULT_MAIL_ATTRIBUTE	The default identity store attribute used to find the e-mail addresses of the recipients. The default value is MX_MAIL_PRIMARY.
DEFAULT_MAIL_LANGUAGE	If no language is set on the user, or the user specifies a language that does not have a template, this language is used.
DEFAULT_MAIL_LANGUAGE_ATTRIBUTE	The default identity store attribute containing the language setting for the user.
MAIL_DEBUG	Turning debug mode on or off. 1 (or TRUE) means that debug is turned on, and all e-mails are sent to the MAIL_DEBUG_RECIPIENTS.
MAIL_DEBUG_RECIPIENTS	E-mail address used for all messages when MAIL_DEBUG is enabled.

Package Constant	Description
MAIL_LANGUAGEATTRIBUTE	The identity store attribute containing the language setting for the user.
MAIL_ORIGINATOR	E-mail address used as originator in messages sent by the IdM system
MAIL_SMTP_CONFIGURATION	Defines a set of properties used by JavaMail API to connect to the SMTP server. It takes one of the following values: - SMTP - SMTPS - SMTP_TLS <custom value> The format of the custom value is: property1=value1, property2=value2. For more information, see uSendMailAuth [page 1216]. This constant is used only if there is a specified value for MAIL_SMTP_USERNAME constant.
MAIL_SMTP_HOST	IP address or hostname of the SMTP server
MAIL_SMTP_USERNAME	Username used to authenticate to the SMTP server. Depending on the specified value, there are the following options: - Specified value If a value is specified, uSendMailAuth function is called in sap_sendNotification script to send an e-mail with authentication (username and password) to the mail server. In that case, the MAIL_SMTP_PASSWORD and MAIL_SMTP_CONFIGURATION are mandatory. - No specified value If there is no value specified, the e-mail to the mail server is sent without authentication. In that case uSendMail function is called. For more information, see uSendMail [page 1214].
MAIL_SMTP_PASSWORD	Password used to authenticate to the SMTP server. It is used only if there is a specified value for MAIL_SMTP_USERNAME constant.
MAIL_SMTP_PORT	Port number for the SMTP service, if not given 25 is used.

Package Constant	Description
MX_GUI_URL_PREFIX	The URL of the WebUI This constant is used to generate the URL for an approval in the notification process. It has one of the forms: • <code>http(s)://<host or ip>:<port>/</code> • <code>http(s)://<host or ip>:<port>/idm</code> • <code>http(s)://<host or ip>:<port>/idm/</code>

If you want to trigger a notification as a result of an SAP Provisioning Framework process, in the Identity Management Administration User Interface, navigate to ► [System Configuration](#) ► [Packages](#) ► [com.sap.idm.util.notification](#) ► [Constants](#) ► and specify the value of the appropriate NOTIFYEVENT constant to point to the template you want to use:

Package Constant	Value	Comment
NOTIFYEVENT_ASSIGNMENT_COMPLETED	PF Privilege assignment notification	The mail template used for completed assignment event. The value defined for the constant is a unique name (ID) for the notification template and needs to match the name defined for this template on the Message Templates tab. You can copy this name from the Message Templates tab in the Identity Management Administration User Interface. Select the MX_SAP_PF category, choose Go and copy the name of the message template that you want to use. For more information, see: <i>Maintaining Message Templates</i>.

Package Constant	Value	Comment
NOTIFYEVENT_ASSIGNMENT_FAILED	PF Assignment failed notification	The mail template used for assignment failed event. The value defined for the constant is a unique name (ID) for the notification template and needs to match the name defined for this template on the Message Templates tab. You can copy this name from the Message Templates tab in the Identity Management Administration User Interface. Select the MX_SAP_PF category, choose Go and copy the name of the message template that you want to use. For more information, see: <i>Maintaining Message Templates</i>.
NOTIFYEVENT_ASSIGNMENT_REVOKED	PF Privilege removal notification	The mail template used for revoked assignment event. The value defined for the constant is a unique name (ID) for the notification template and needs to match the name defined for this template on the Message Templates tab. You can copy this name from the Message Templates tab in the Identity Management Administration User Interface. Select the MX_SAP_PF category, choose Go and copy the name of the message template that you want to use. For more information, see: <i>Maintaining Message Templates</i>.

Package Constant	Value	Comment
NOTIFYEVENT_PASSWORD_CHANGED	PF Changed password notification	The mail template used for password changed event. The value defined for the constant is a unique name (ID) for the notification template and needs to match the name defined for this template on the Message Templates tab. You can copy this name from the Message Templates tab in the Identity Management Administration User Interface. Select the MX_SAP_PF category, choose Go and copy the name of the message template that you want to use. For more information, see: <i>Maintaining Message Templates</i>.
NOTIFYEVENT_USER_MODIFIED	PF Modified user notification	The mail template used for user modified event. The value defined for the constant is a unique name (ID) for the notification template and needs to match the name defined for this template on the Message Templates tab. You can copy this name from the Message Templates tab in the Identity Management Administration User Interface. Select the MX_SAP_PF category, choose Go and copy the name of the message template that you want to use. For more information, see: <i>Maintaining Message Templates</i>.

Package Constant	Value	Comment
NOTIFYEVENT_USERACCOUNT_CREATED	PF Created user notification	The mail template used for user account created event. The value defined for the constant is a unique name (ID) for the notification template and needs to match the name defined for this template on the Message Templates tab. You can copy this name from the Message Templates tab in the Identity Management Administration User Interface. Select the MX_SAP_PF category, choose Go and copy the name of the message template that you want to use. For more information, see: <i>Maintaining Message Templates</i>.
NOTIFYEVENT_USERACCOUNT_DELETED	PF Deleted user notification	The mail template used for user account deleted event. The value defined for the constant is a unique name (ID) for the notification template and needs to match the name defined for this template on the Message Templates tab. You can copy this name from the Message Templates tab in the Identity Management Administration User Interface. Select the MX_SAP_PF category, choose Go and copy the name of the message template that you want to use. For more information, see: <i>Maintaining Message Templates</i>.

Package Constant	Value	Comment
NOTIFYEVENT_USERACCOUNT_DISABLED	PF Disabled user notification	The mail template used for user account disabled event. The value defined for the constant is a unique name (ID) for the notification template and needs to match the name defined for this template on the Message Templates tab. You can copy this name from the Message Templates tab in the Identity Management Administration User Interface. Select the MX_SAP_PF category, choose Go and copy the name of the message template that you want to use. For more information, see: <i>Maintaining Message Templates</i>.
NOTIFYEVENT_USERACCOUNT_ENABLED	PF Enabled user notification	The mail template used for user account enabled event. The value defined for the constant is a unique name (ID) for the notification template and needs to match the name defined for this template on the Message Templates tab. You can copy this name from the Message Templates tab in the Identity Management Administration User Interface. Select the MX_SAP_PF category, choose Go and copy the name of the message template that you want to use. For more information, see: <i>Maintaining Message Templates</i>.

Notification Process

This package contains a notification process for sending e-mail notifications referenced from an approval or attestation task.

Notification Templates

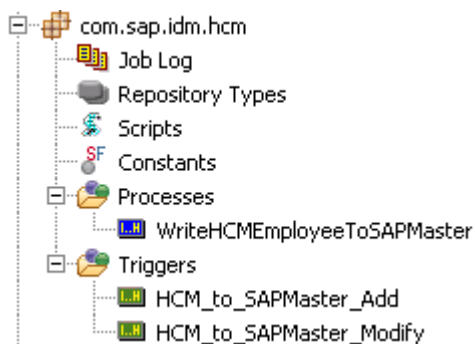
This package contains three groups of notification templates:

- MX_APPROVALS
 - Approval delegation notification
 - Assignment escalation escalated
 - Approval initial notification
 - Approval reminder notification
 - Assignment completed notification
- MX_ATTESTATION
 - Attestation delegation notification
 - Attestation initial notification
 - Attestation reminder notification
- MX_SAP_PF
 - PF Assignment failed notification
 - PF Changed password notification
 - PF Created user notification
 - PF Deleted user notification
 - PF Disabled user notification
 - PF Enabled user notification
 - PF Modified user notification
 - PF Privilege assignment notification
 - PF Privilege removal notification
 - PF Role assignment notification
 - PF Role removal notification

3.2.4.18 HCM Integration Package

The HCM integration package [com.sap.idm.hcm](#) contains the processes, constants and scripts required for SAP HCM integration.

The structure of this package is as follows:



Triggers

The HCM Integration package contains a set of triggers:

- HCM_to_SAPMaster_Add
- HCM_to_SAPMaster_Modify

Processes

Along with the triggers, this package contains the following process: WriteHCMEmployeeToSAPMaster.

Package Constants

The following package constants are defined:

Package Constant	Description
HR_STAGING_AREA_IDS_ID	Id of the staging area identity store
SAP_MASTER_IDS_ID	Id of the master identity store
SAP_SPML_STAGING_AREA	Id of the SPML staging area identity store

Along with the processes, this package contains several scripts used in various cases, including importing phone numbers, initializing passwords, and checking whether the employee is a manager.

3.2.4.19 Reporting Package

The reporting package [com.sap.idm.reporting](#) contains regular processes and forms required for configuring standard reports.

The structure of this package is as follows:

- ▼  com.sap.idm.reporting
 -  Job Log
 - >  Repository Types
 -  Scripts
 -  Constants
 - ▼  Processes
 -  UserHistoryAssignmentsProcess
 -  UserHistoryAttributesProcess
 -  AssignedUsersForRoleProcess
 -  AssignedUsersForPrivilegeProcess
 -  UsersAttributesProcess
 -  UsersAssignmentsProcess
 -  DoublePrivilegeAssignmentsProcess
 - ▼  Forms
 - ▼  Reports
 - ▼  Self-Services Forms
 -  DoublePrivilegeAssignmentsReport
 -  UsersAssignmentsReport
 -  UsersAttributesReport
 -  UserHistoryAssignmentsReport
 -  UserHistoryAttributesReport
 -  AssignedUsersForRoleReport
 -  AssignedUsersForPrivilegeReport
 - >  Jobs

Processes

The package contains a number of regular processes. Each process in this package contains an action task with a To Identity Store pass and a specific script `sap_generate<report_name>Report`. The script defines the SQL select query for a given report, that is, what data will be extracted in the report table.

Forms

The package contains a number of report forms which are executed in the Identity Management User Interface. Each process delivered with this package is referenced on the [Result Handling](#) tab of the corresponding form.

Related Information

[Standard Reports \[page 780\]](#)

3.2.5 Importing the Provisioning Framework for SAP Identity Management 8.0

Prerequisites

- You have installed the SAP Identity Management Developer Studio and performed the initial configuration.
- You have created an identity store to use with the provisioning framework for SAP Identity Management 8.0.
- You are familiar with the dependencies between packages. For more information, see [Package Dependencies \[page 404\]](#).
- You have created a dispatcher for running jobs.

Context

The provisioning framework for SAP Identity Management 8.0 is distributed in packages. Each package has a specific purpose which its name reflects. There is an engine package, a package for a specific connector, a package for notifications, a package for user interface forms, and a custom package. To have a successful provisioning, you need to import an engine package, a custom package, and at least one connector package into the Identity Management Developer Studio.

Note

You cannot import the SAP provisioning engine and custom packages into multiple identity stores. Thus, any connector package that depends on them (see *Package Dependencies*) cannot be imported into multiple identity stores. If you need to do that, you can import them under different names but then you need to manually change all package dependencies. Alternatively, you can import the provisioning framework from version 7.2.

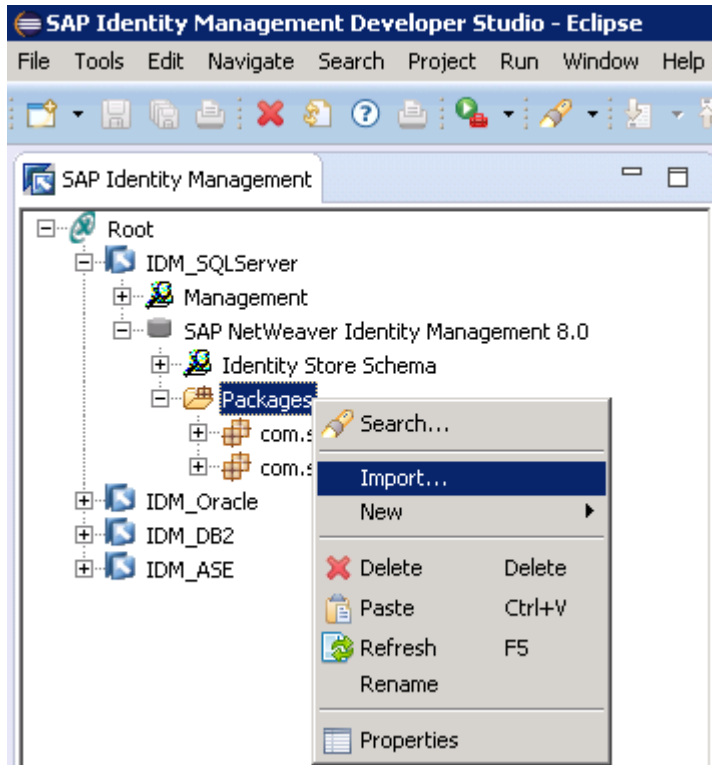
Caution

If you have implemented package customizations (for example, in packages imported under new names, package copies or the custom package), note that when you import a new version of SAP delivered packages, you should manually apply the changes in your customized packages.

To import a package, proceed as follows:

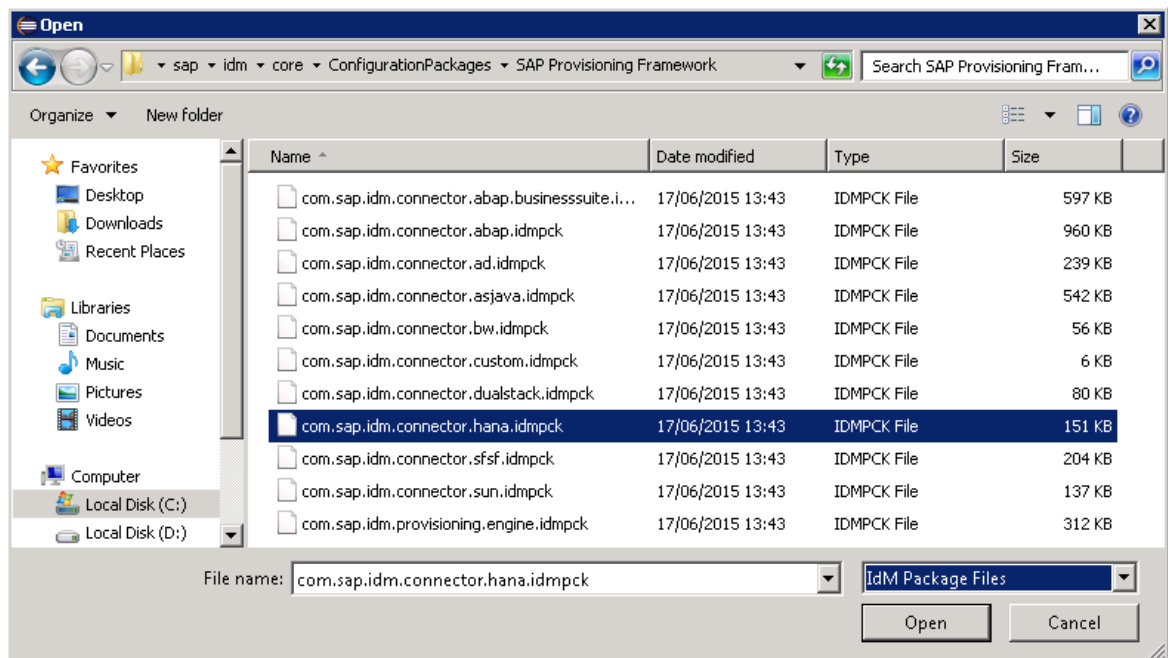
Procedure

1. In the Identity Management Developer Studio tree view, select the [Packages](#) folder of your identity store and choose [Import](#) from the context menu.



2. Navigate to the location of the package file, select it and choose [Open](#). For more information, see [Package Files \[page 399\]](#).

For example, select the HANA connector package file [com.sap.idm.connector.hana.idmpck](#).



3. In the [Enter import reason](#) dialog box, configure the following:
 - (Optional) Enter a reason for the import.
 - Select [Import](#).

The package is imported with its package qualified name.

- Select *Import as new package* and enter a name for the package.
The package is imported under a new name. You can use this option to modify an SAP provisioning framework package and keep your changes while updating the provisioning framework.

Enter import reason

Package Name:

Export Date:

Description:

Import Version:

Current Version:

Reason for Import:

☒ Import

☐ Import as new package

com.bestrun.

OK Cancel

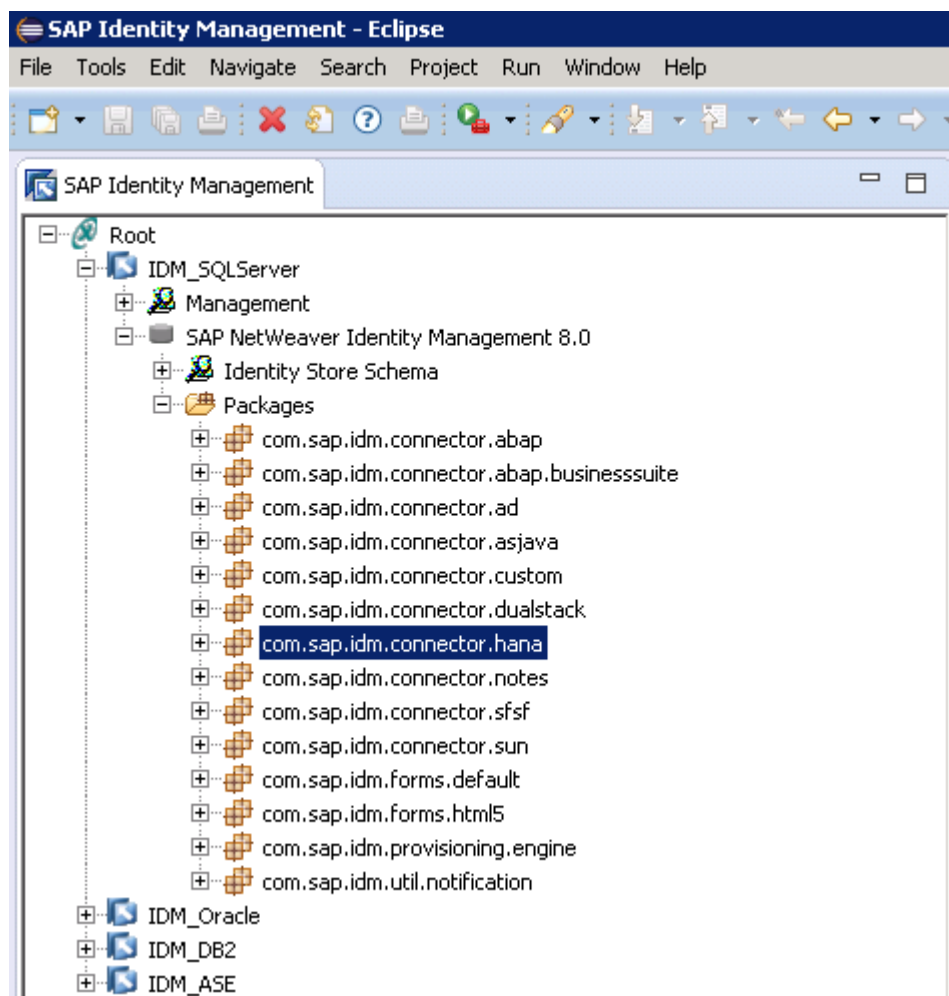
Note

Do not import the engine package as a new package. We recommend that you do not rename the engine package while importing it. If you import it under a new name, the connectors will not function as intended.

4. Choose *OK*.

Results

The package is imported into the Identity Management Developer Studio.



The *Status* on the *Dependencies* tab of the imported package should be *Linked OK* or *Linked to different version* (Indicates that objects in the imported packages are linked to different package version).

Related Information

[Installing the Identity Management Developer Studio](#)

[Adding the Initial Identity Store](#)

[Package Dependencies \[page 404\]](#)

[Creating and Managing the Dispatcher\(s\)](#)

[Packages \[page 91\]](#)

[Renaming Packages \[page 206\]](#)

[Maintaining Repositories \[page 492\]](#)

[Changing Repository Type of Repositories \[page 499\]](#)

3.2.6 Updating the Provisioning Framework

Context

If you want to update the provisioning framework for SAP Identity Management 8.0 (distributed in packages), then follow the instructions for importing the framework.

As updating the provisioning framework overwrites the existing framework, we do not recommend changing the framework itself. Instead, import the packages that you want to modify under a new name and make the changes there or use the custom package for your customizations.

⚠ Caution

If you have implemented package customizations (for example, in packages imported under new names, package copies or the custom package), note that when you import a new version of SAP delivered packages, you should manually apply the changes in your customized packages.

ℹ Note

The engine package is an SAP read-only package and thus it cannot be modified.

To update the provisioning framework for SAP Identity Management 8.0, proceed as follows:

Procedure

1. In the Identity Management Developer Studio tree view, select the *Packages* folder of your identity store and choose *Import* from the context menu.
2. Select the package file from the file system and choose *Open*. You can find it in the `<Install_folder>\Core\ConfigurationPackages\SAP Provisioning Framework` folder.
3. In the *Enter Import Reason* dialog box, you can do the following:
 - (Optional) Enter a reason for the import.
 - Choose *Import*.

If you have modified the previous version of this package, the import of the new version will overwrite your changes. If your changes to the configuration are overwritten, you have to make them again manually.

- Choose *Import as New Package* and enter a name for the package.
If you choose *Import as New Package*, you will import the package under a new name.

Note

Do not import the engine package as a new package. We recommend that you do not rename the engine package while importing it. If you import it under a new name, the connectors will not function as intended.

4. Choose *Save*.

3.3 Maintaining Repositories

You can create, enable, disable, delete, copy, import, export, and change the repository type of the repositories, using the Identity Management Administration User Interface.

Note

To be able to perform these actions, you need correct authorizations for the Identity Management Administration User Interface. See section *Access to the Identity Management User Interfaces (URLs)* in *SAP Identity Management Security Guide* for authorization details and *Adding User(s) to the Identity Store* in *Installing and Configuring SAP Identity Management User Interface*.

For more information about migrating existing 7.2 repository types to 8.0, see [Changing Repository Type of Repositories \[page 499\]](#).

Related Information

[Access to the Identity Management User Interfaces \(URLs\)](#)

[Adding User\(s\) to the Identity Store](#)

3.3.1 Creating Repositories

You can create a repository in the SAP Identity Management Administration User Interface for each system in the system landscape. The repository data provides the connection information to the system and other system-specific information.

Prerequisites

- The repository type for a repository is created in the Identity Management Developer Studio.

Context

A repository is based on a repository type that is configured with a set of constants, both repository and repository type specific. Some of these constants have a default value that you can modify. For other constants you need to provide the value when you add a repository of this type. See *Modifying Repository Constants* for details.

To create a repository in the SAP Identity Management Administration User Interface, proceed as follows:

Procedure

1. Log on to SAP Identity Management Administration User Interface at `http://<host>:<port>/idm/admin`.

2. Select the *System Configuration* tab and then *Repositories* from the menu on the left.
3. Choose *Create* and select *Create New Repository*.
4. A *Create Repository* dialog box appears. Enter a unique name and a description for the new repository.

Note

The name can contain alphanumeric characters only (A-Z, a-z, 0-9). Spaces or special characters are not supported. When using Oracle, the maximum length of the name of the repository is six characters.

5. In *Create Repository* dialog box, select a repository type, for example:
 - SAPHANA
 - GRC10
 - ASJavaDB
 - ASJavaLDAP
 - ActiveDirectory
 - SFSF
6. Choose *OK*.
7. Specify the repository constants that apply to the system type. See *Appendix A: Repository Constants* for a list of constants per repository type. This appendix also contains additional information, for example, that using SSL is a prerequisite for password provisioning.

The value of the repository constant SYSTEM_PRIVILEGE is automatically set to PRIV:SYSTEM:<REP_NAME>, where REP_NAME is the name of the repository. This value cannot be changed.

Note

The user you create with a repository constant should be the same as the one in the system (AS Java, AS ABAP, LDAP, etc) you are connected to, which means that the repository and the system should have the same user. In addition, the user name defined by the repository constant must match with the user's logon ID in the Administration User Interface. That means that the logon ID of the SAP Identity Management maps to the user name in the repository.

Note

If you are setting up any SAP Business Suite systems and have provisioning tasks where a corresponding user account should not be created or maintained, then you must set up an additional repository for the system that contains the NO_USER_ACCOUNT constant with the value 1.

Caution

If the repository constants apply to an AS ABAP Release 4.6C, make sure you enter the password to use for the connection in upper case.

Note

If you delete a repository type, the repositories of this type become so called orphaned repositories. If you then recreate a repository type, with the same name as the deleted one, the orphaned repositories become repositories of that type again (no longer orphaned).

Related Information

[Appendix A: Repository Constants](#)

3.3.2 Enabling Repositories

When you enable a disabled repository, you start the provisioning to the corresponding target system. All scheduled jobs to the target system start automatically.

Prerequisites

- The repository that you want to enable has been disabled.

Context

To enable a repository, proceed as follows:

Procedure

1. Log on to SAP Identity Management Administration User Interface at `http://<host>:<port>/idm/admin`.
2. Select the *System Configuration* tab and then *Repositories* from the menu on the left.
3. Select a repository from the list and choose *Enable*.
4. (Optionally) Choose the refresh button to the right of the screen area to see the state of the repository.

Results

The repository is enabled. To view the history of enabled or disabled changes for repositories, choose  *Configuration History*  *Repository Operations* .

3.3.3 Disabling Repositories

When you disable a repository, you stop the provisioning to the corresponding target system.

Context

When you disable a repository, you stop the repository jobs for this repository and the provisioning to the corresponding target system. All repository jobs for the disabled repository stop and wait in the provisioning queue.

However, if you run a regular job with a pass that references a disabled repository, the job is executed, but no provisioning is triggered. The provisioning to the target system starts once the repository is enabled again.

Caution

Make sure that there are no tasks and jobs running for a repository when you disable it. Note also that the provisioning queue might fill up with entries waiting to be processed if tasks and jobs are triggered to a repository while disabled.

Disabling a repository is relevant in the following use cases:

- You want to change the repository type of a given repository. In this case, disabling the repository is a mandatory step.
- There is no connection to the target system, or it is offline for maintenance. Disabling the repository in this case stops the provisioning to the corresponding system and prevents jobs from failing.

To disable a repository, proceed as follows:

Procedure

1. Log on to SAP Identity Management Administration User Interface at `http://<host>:<port>/idm/admin`.
2. Select the *System Configuration* tab and then *Repositories* from the menu on the left.
3. Select a repository from the list and choose *Disable*.
4. (Optionally) Choose the refresh button to the right of the screen area to see the state of the repository.

Results

The repository is disabled. To view the history of enabled or disabled changes for repositories, choose

 *Configuration History*  *Repository Operations* .

3.3.4 Copying Repositories

Context

To create a copy of a repository in the SAP Identity Management Administration User Interface:

Procedure

1. Log on to SAP Identity Management Administration User Interface, accessible from `http://<host>:<port>/idm/admin`.
2. Select the *System Configuration* tab and then *Repositories* from the menu on the left.
3. Select a repository from the list and choose *Copy Repository*.
4. A *Copy Repository* dialog box appears. Enter a name and a description of the new, copied repository.
5. Choose *OK*.

3.3.5 Exporting Repositories

Context

To export a repository in the SAP Identity Management Administration User Interface:

Procedure

1. Log on to SAP Identity Management Administration User Interface, accessible from `http://<host>:<port>/idm/admin`.
2. Select the *System Configuration* tab and then *Repositories* from the menu on the left.
3. Choose *Export*, and then one of the following export options:
 - *All Repositories*
 - *Repositories in Table*

The repositories are exported as a CSV file.

The CSV file holds the following information:

- repository name
- description
- ID of the repository type
- enabled/disabled status
- qualified name of the repository type

In addition, the file specifies how constants are transported. For example: Boolean is transported as *true* or *false*, an entry reference as *MSKEYVALUE* or *MSKEY* of the entry, the *MX_PRIV_GROUPING_RULE* constant as a value ranging from *P:0* to *P:7*, an LDAP or a JDBC URL as URLs.

Constants with encrypted values are exported as empty values.

ⓘ Note

For security reasons, all repository constants with encrypted values, like passwords, are exported as empty values.

3.3.6 Importing Repositories

Context

To import a repository in the SAP Identity Management Administration User Interface:

Procedure

1. Log on to SAP Identity Management Administration User Interface, accessible from `http://<host>:<port>/idm/admin`.
2. Select the *System Configuration* tab and then *Repositories* from the menu on the left.
3. Choose *Import*.
4. Select the CSV file containing the repositories to import.

ⓘ Note

If an existing repository has the same name as the imported one, you can select the *Overwrite Existing Repositories* option to have the existing repository updated with the values from the imported file.

If the imported file contains encrypted repository constants, the existing repository constants are overwritten.

If the imported file contains encrypted repository constants with empty value, the existing repository constants are not overwritten.

Note

For security reasons, if the CSV file contains passwords in plain text, they are encrypted when the repositories are imported.

5. To start the import, choose [Upload](#).

3.3.7 Deleting Repositories

Context

To delete a repository in the SAP Identity Management Administration User Interface:

Procedure

1. Log on to SAP Identity Management Administration User Interface, accessible from `http://<host>:<port>/idm/admin`.
2. Select the [System Configuration](#) tab and then [Repositories](#) from the menu on the left.
3. Select a repository from the list and choose [Delete](#).

Note

A confirmation dialog box appears every time you are deleting a repository. Note the implications before you confirm the deletion.

3.3.8 Changing Repository Type of Repositories

You can change the repository type of a given repository in the SAP Identity Management Administration User Interface.

Prerequisites

- The repository that you want to change the repository type for is fully operational, meaning the repository constants are configured, the Initial Load job run successfully and the provisioning to the target system is working.
- The repository is disabled.

- The new repository type is available in the SAP Identity Management Developer Studio.

Context

Changing the repository type of a repository is relevant in the following use cases:

- You have upgraded from SAP Identity Management 7.2 to 8.0 and you want to use the Provisioning framework for SAP Identity Management 8.0. In this case, you can configure your 7.2 repositories to change their repository types to the ones delivered with the new Provisioning framework. All customizations required for the business logic of the repository processes or the repository type implementation should be implemented manually in the relevant new 8.0 connector.

❖ Example

For example, you can change the repository type of a 7.2 JAVA repository to be the ASJavaDB repository type.

- You have created a custom repository type with custom specific features and want to change the repository type of a given repository to be the custom one.

Procedure

1. Log on to SAP Identity Management Administration User Interface at `http://<host>:<port>/idm/admin`.
2. Select the *System Configuration* tab and then *Repositories* from the menu on the left.
3. Select a repository that has already been disabled and choose *Change Repository Type*.
4. In the *Change Repository Type* dialog, make the required entries:
 - a. Select a repository type.
 - b. (Optionally) Enter a description.
 - c. Enter a name for the existing (old) repository. By default, the suffix **_OLD** is added to the name of the existing (old) repository. You can overwrite this with a new name of your choosing.

📘 Note

The name can contain alphanumeric characters only (A-Z, a-z, 0-9). Spaces or special characters are not supported. When using Oracle, the maximum length of the repository name is six characters.

5. Choose *OK*.
6. Check the repository constants and adjust their values if necessary.

📘 Note

If the existing (old) repository and the one with the changed repository type have the same repository constants, their values are entered automatically. For other repository constants, you need to enter their values manually.

7. Enable the repository.

Results

The repository changes its repository type but keeps the same name and repository constants configuration. The existing (old) repository is disabled. It can only be enabled again if jobs are scheduled to run on it. If not, you might want to delete it.

Note

Since you cannot change the repository type of a repository job, if you want to run the job with repository that has changed its repository type, you need to create the repository job again and set the respective repository type.

To view the history of the repository type change for the repository, choose ► [Configuration History](#) ► [Repository Operations](#) ►.

To view the history of the repository constants affected by the repository type change for the repository, choose ► [Configuration History](#) ► [Repository Constants](#) ►.

Related Information

[Repository Constants \[page 1234\]](#)

3.4 Maintaining Identity Store Entries

This section provides information about maintaining identity store entries, such as users, roles, privileges and dynamic groups in the SAP Identity Management User Interface.

Prerequisites

- You need to have a working Identity Management system with the necessary access rights for the Identity Management User Interface.
- You have imported the forms package `com.sap.idm.forms.default` into the Identity Management Developer Studio.

You can maintain identity store entries by using forms in the Identity Management User Interface. A set of default forms for the Identity Management User Interface is delivered as a separate package (`com.sap.idm.forms.default`) with the Provisioning Framework for SAP Identity Management 8.0. Forms are configured in the Identity Management Developer Studio.

For more information about the functionality of the User Interface, the content of the forms package and how to import it, see the *Related Information* section.

The following sections describe how to maintain identity store entries in the Identity Management User Interface by using the default forms delivered with the forms package. The Identity Management User Interface is accessible from `http://<host>:<port>/idm`.

Related Information

[Identity Management User Interface \[page 76\]](#)

[Forms Package \[page 473\]](#)

[Importing the Provisioning Framework for SAP Identity Management 8.0 \[page 487\]](#)

[Viewing Attribute History \[page 551\]](#)

3.4.1 Maintaining Identities

You can display, search for and maintain identities (users) in the Identity Management User Interface by using the default forms delivered with the `com.sap.idm.forms.default` package.

Related Information

[Creating Identities \[page 503\]](#)

[Modifying Identities \[page 504\]](#)

[Enabling or Disabling Account of Identities \[page 505\]](#)

[Deleting Identities \[page 506\]](#)

[Viewing Attribute History \[page 551\]](#)

3.4.1.1 Creating Identities

Context

To create an identity (user) in the Identity Management User Interface, proceed as follows:

Procedure

1. On the *Manage* tab, select *Person* from the *Show* list and choose *Create*.
2. In the *Choose Task* dialog box, choose ► *Identity* ► *Create Identity* ► *Choose Task* ►.
3. Select a tab and provide the necessary information:

Tab	Description
<i>General Attributes</i>	Enter the required information.
<i>Account Attributes</i>	Provide the following information: • <i>Account Status</i> <i>Restricted User</i> - Select the checkbox if the user is a restricted database user in SAP HANA. Specify the time when the entry is valid from and the time when the entry is no longer valid. • <i>Authentication</i> Choose an authentication method.
<i>Assigned Roles and Privileges</i>	Search for and select the role or the privilege you want to assign to the user.
<i>Settings</i>	Enter the necessary information.

4. Choose *Save*.

3.4.1.2 Modifying Identities

Context

To modify an identity (user) in the Identity Management User Interface:

Procedure

1. On the [Manage](#) tab, select [Person](#) from the [Show](#) list.
2. In the [Find](#) field, enter the name of the user you want to modify and choose [Go](#).
3. Select the user and choose [Choose Task](#).
4. In the [Choose Task](#) dialog box, choose [Identity](#) > [Change Identity](#) > [Choose Task](#) .
5. Select a tab and change the information about the user.
6. Choose [Save](#).

3.4.1.3 Assigning Privileges, Roles and Groups to Identities

Context

To assign privileges, roles and/or groups to an identity (user) in the Identity Management User Interface:

Procedure

1. On the [Manage](#) tab, select [Person](#) from the [Show](#) list.
2. In the [Find](#) field, enter the name of the user and choose [Go](#).
3. Select the user and choose [Choose Task](#).
4. In the [Choose Task](#) dialog box, choose [Identity](#) > [Assign Privileges, Roles and Groups](#) > [Choose Task](#) .
5. You can do the following:
 - To assign a role and/or a privilege to the user, select the [Assigned Roles and Privileges](#) tab. Search for the role and/or the privilege you want to assign to the user, select it and choose [Add](#). In the [New Assignment](#) dialog box, specify the [Direct Validity Assignment](#) and choose [Add](#).

📘 Note

When assigning a role to a user, all privileges of this role are automatically assigned to the user.

📘 Note

The status of the role depends on the status of the assigned privileges to this role. For example:

- The status of the role is *OK*, although the status of the assigned privileges to this role is *OK* or *Pending*.
- The status of the role is *Failed*, if at least one of the assigned privileges to this role is *Failed*.

This is a standard behavior of SAP Identity Management. For all use cases, see SAP Note [2105550](#).

- To assign a group to the user, i.e. to make the user a member of a group, select the *Assigned Groups* tab. Search for the group you want to assign to the user, select it and choose *Add*.

6. Choose *Save*.

3.4.1.4 Enabling or Disabling Account of Identities

Context

When you disable an identity (user) in SAP Identity Management User Interface, its user account is locked in SAP Identity Management and all target systems of this identity. As a result, the user is not able to logon to SAP Identity Management User Interface and the target systems.

To disable or enable the account of an identity (user) in the Identity Management User Interface, proceed as follows:

Procedure

1. On the *Manage* tab, select *Person* from the *Show* list.
2. In the *Find* field, enter the name of the user and choose *Go*.
3. Select the user and choose *Choose Task*.
4. In the *Choose Task* dialog box, choose ► *Identity* ► *Enable/Disable Identity* ► *Choose Task* ⌵.
5. Select *Enable/Disable Identity* tab. You can do the following:
 - To disable the account of the user, select *Account is Disabled* checkbox.
 - To enable the account of the user, deselect *Account is Disabled* checkbox.
6. Choose *Save*.

3.4.1.5 Deleting Identities

Context

When you delete an identity (user) in SAP Identity Management User Interface, a deprovisioning process is triggered for all target systems of this identity. Once the identity is deleted from the target systems, it is deleted in SAP Identity Management.

To delete an identity (user) in SAP Identity Management User Interface, proceed as follows:

Procedure

1. On the *Manage* tab, select *Person* from the *Show* list.
2. In the *Find* field, enter the name of the user and choose *Go*.
3. Select the user and choose *Choose Task*.
4. In the *Choose Task* dialog box, choose *Identity* > *Delete Identity* > *Choose Task*.
5. Choose *Delete*.

Note

It is generally not recommended to delete entries in the identity store:

- Entries are needed for statistical and audit purposes.
- An employee may return to the organization at a later time, and it simplifies the process if the person entry can be re-used.

Instead it is recommended to set the entry as inactive.

3.4.2 Maintaining Roles

Business roles represent the business tasks of an employee, for example: manager, accounting, employee and others. Business roles help manage and structure the assignment of technical roles (privileges) in the target systems. For more information, see [Defining a Role Model \[page 21\]](#).

You can display, search for and maintain business roles in the Identity Management User Interface by using the default forms delivered with the `com.sap.idm.forms.default` package.

When assigning a role to a user, all privileges of this role are automatically assigned to the user, as are any privileges of any child roles.

Note

The status of the role depends on the status of the assigned privileges to this role. For example:

- The status of the role is *OK*, although the status of the assigned privileges to this role is *OK* or *Pending*.
- The status of the role is *Failed*, if at least one of the assigned privileges to this role is *Failed*.

This is a standard behavior of SAP Identity Management. For all use cases, see SAP Note [2105550](#) .

By default, identity store entries of all entry reference attributes (for example, all attributes starting with MXMEMBER and MXREF) are sorted out alphabetically by their DISPLAYNAME in Identity Management User Interface (UI). This applies for displaying entries (such as roles, privileges or dynamic groups) in the [Details](#) pane of the UI or viewing the assigned roles, privileges or dynamic groups of a user.

Related Information

[Creating Roles \[page 507\]](#)

[Modifying Roles \[page 521\]](#)

[Deleting Roles \[page 521\]](#)

3.4.2.1 Creating Roles

Context

To create a role in the Identity Management User Interface:

Procedure

1. On the *Manage* tab, select *Role* from the *Show* list and choose *Create*.
2. In the *Choose Task* dialog box, choose  *Role*  *Create Role*  *Choose Task* .
3. Select a tab to enter information about the role.

Related Information

[Specifying the General Properties for a Role \[page 508\]](#)

[Specifying Member Events for a Role \[page 509\]](#)

[Specifying Approvers for a Role \[page 515\]](#)

[Specifying Visibility of a Role \[page 515\]](#)

[Specifying Restrictions for a Role \[page 516\]](#)

[Assigning Privileges or Dynamic Groups to a Role \[page 517\]](#)

[Configuring Contexts for a Role \[page 518\]](#)

[Viewing Users Assigned to a Role \[page 519\]](#)

[Building a Role Hierarchy \[page 519\]](#)

[Specifying Attester for Role Assignment \[page 520\]](#)

3.4.2.1.1 Specifying the General Properties for a Role

When you configure a role, you use the *General* tab to specify the general properties of the role.

Context

To specify the general properties of the role:

Procedure

1. Select the *General* tab.
2. Enter the required information:

Field	Description
<i>Unique ID</i>	Enter a name for the role. This name must be unique within the identity store (MSKEYVALUE). The recommended syntax is: ROLE:<Role name>
<i>Display Name</i>	Enter a display name for the role that is to be used in the Identity Management User Interface.
<i>Description</i>	Enter a text describing the role.

3. Select another tab to specify other properties of the role or choose *Save*.

3.4.2.1.2 Specifying Member Events for a Role

When you configure a role, you use the [Member Events](#) tab to specify the member event handling for the role.

Context

For more information about processes that are executed when a role is assigned to or removed from an entry, see [Member Event Handling \[page 81\]](#)

To specify the member event handling for the role, proceed as follows:

Procedure

1. Select the [Member Events](#) tab.
2. Provide the necessary information in the following fields:

Note

Although it is possible to specify here each process listed below, it is recommended that the process is defined on the repository type of the referenced repository as the same process can be used for several roles/privileges referencing this process. In this case, the process here is inherited from the repository type of the referenced repository.

Alternatively, if you specify a Validate process containing an approval task on a child role with assigned privileges, be aware that this could lead to unexpected results when assigning or de-assigning the parent role.

In case a Validate Remove process containing an approval task is defined on a child role with assigned privileges, expect the following: a removal of the parent role from the user will result in de-assigning of the privileges even though you decline the Validate Removal process on the child role.

Field	Description
Repository name	Select the repository that is specified for the role.

Field	Description
Validate Add Process	Select the process that should be used to verify that the add operation is allowed for the given entry. You can select a process of type Regular or Validate Add Process. This can for example be an approval or a call to GRC Access Control. This process uses the repository specified for the Validate processes. If no repository is specified, the default repository will be used. This process returns either true or false. If this is a process containing an approval task, this is handled automatically, but if you are using a regular process, for instance to check a group membership in an external system, this must be done manually by setting MX_ATTR_STATE to 1 to allow the assignment and 3 to reject it. If this process is defined, the Add Member Process will not be performed unless this process completes successfully. Select a value in the When to Execute Validate Task drop-down list to specify when the process should be executed relative to the valid from and valid to specified for the assignment. You can choose between: • When Valid - Execute the process at the "Valid from" specified in the assignment request. • Immediately - Execute the process when the assignment is requested. This is the default value for validate processes. Note This option works only for direct assignments of privileges. It does not work for privileges assigned through roles. Assignments of roles containing privileges are made as soon as the valid from date is reached, then the role and the privilege will be assigned. • Number of days - Select a number of days before the "Valid from" when the process should be executed. If you select 5 days before, the process will be executed 5 days before the date specified as "Valid from" in the assignment request.

Field	Description
Validate Remove Process	Select the process that should be used to verify that the remove operation is allowed for the given entry. You can select a process of type Regular or Validate Remove Process. This can for example be an approval or a call to GRC Access Control.  Note If a role assignment has expired, it is not possible to approve it. That is, you cannot select a process containing an approval task in the Validate Remove Process field. This process uses the repository specified for the Validate processes. If no repository is specified, the default repository will be used. This process returns either true or false. If this is a process containing an approval task, this is handled automatically, but if you are using a regular process, for instance to check a group membership in an external system, this must be done manually by setting MX_ATTR_STATE to 1 to allow removing the assignment and 3 to reject it. If this process is defined, the Remove Member Process will not be performed unless this process completes successfully.
Validate Modify Validity Process	Select the process that should be used to verify that the modify validity operation is allowed for the given entry. You can select a process of type Regular or Validate Modify Process. This can for example be an approval or a call to GRC Access Control. This process returns either true or false. If this is a process containing an approval task, this is handled automatically, but if you are using a regular process, for instance to check a group membership in an external system, this must be done manually by setting MX_ATTR_STATE to 1 to allow modifying the validity of the assignment and 3 to reject it. If this process is defined, the Modify Member Process will not be performed unless this process completes successfully.

Field	Description
Add Member Process	Select a process that will be performed on the pending value object created when the role is assigned to an entry. This process will perform its operations, for example request an approval when assigning a role. Depending on the status of the operations, the role will or will not be assigned to the entry. The role is not assigned until the process has completed. You can select a process of type Regular or Add Member Process. Any approvers to the Add Member Process can be added on the Approval tab of an Approval task. This process uses the repository specified above. If no repository is specified, the default repository will be used. Note If a Validate Add Process is defined, the Add Member Process will not be executed unless the Validate Add Process has completed successfully.
Modify Validity Process	Select a process that will be performed on the pending value object created when the validity of an assignment is modified, either extended or shortened. This process will perform its operations, for example updating the validity in the target application. Depending on the status of the operations, the validity will or will not be modified. The validity is not modified until the process has completed. You can select a process of type Regular or Modify Member Process. This process uses the repository specified above. If no repository is specified, the default repository will be used. Note If a Validate Modify Process is defined, this Modify Member Process will not be executed unless the Validate Modify Process has completed successfully.

Field	Description
Remove Member Process	If the Remove Member Process is specified, the role or privilege will not be removed immediately, but a pending value object is created. Select the process to be executed on this object. The process will then for instance revoke the authorizations in the target system before a privilege is physically removed for the object. The role or privilege is not removed until the process has completed. You can select a process of type Regular or Remove Member Process. This process uses the repository specified above. If no repository is specified, the default repository will be used. If a Validate Remove Process is defined, this Remove Member Process will not be executed unless the Validate Remove Process has completed successfully.

Field	Description
Notification Process	Select a process that will be used to send a notification that an assignment is about to expire. This process is normally specified to be executed a number of days before the assignment expires. You can select a process of type Regular or Notify Process. This process may perform actions like: • Send an e-mail to the user that the assignment is about to expire • Extend the assignment by a given period of time • Create an approval to the manager Select a value in the When to Execute Notification Task drop-down list to specify when the process should be executed relative to the valid from and valid to specified for the assignment. You can choose between: • When Valid - Execute the process at the "Valid from" specified in the assignment request. • Immediately - Execute the process when the assignment is requested. This is the default value for validate processes. Note This option works only for direct assignments of privileges. It does not work for privileges assigned through roles. Assignments of roles containing privileges are made as soon as the valid from date is reached, then the role and the privilege will be assigned. • Number of days - Select a number of days before the "Valid from" when the process should be executed. If you select 5 days before, the process will be executed 5 days before the date specified as "Valid from" in the assignment request.

3. Select another tab to specify other properties of the role or choose [Save](#).

3.4.2.1.3 Specifying Approvers for a Role

When you configure a role, you use the [Approvers](#) tabs to specify approvers for the member event handling.

Context

To specify an approver:

Procedure

1. Select the [Approvers](#) tab.
2. Search for an approver.
3. You can do the following:
 - To add an approver, select the approver and choose [Add](#).
 - To delete an approver, select the already assigned approver and choose [Delete](#).

Note

The default number of escalation levels possible after an approval has timed out is three. You can specify escalation approvers defined for each escalation level following the above procedure.

4. Select another tab to specify other properties of the role or choose [Save](#).

Related Information

[Member Event Handling \[page 81\]](#)

3.4.2.1.4 Specifying Visibility of a Role

When you configure a role, you use the [Visibility](#) tab to specify who should be allowed to see the role. You can also define an owner of the role.

Context

The visibility constraint is used in the following cases:

- When searching for entries to add to the role. Only entries which the user has access to are visible in the search result.

- When managing the role. The role members are only visible if the user has access to them.
- When assigning a role to a user, only roles that the user has access to are visible.

When you specify the visibility of a role, you add the entries that should be defined as entry owners. This can either be a user (entry), a privilege, a role or a dynamic group. If the owner is a privilege, role or dynamic group, all their members are considered as owner entries.

To add an entry owner:

Procedure

1. Select the [Visibility](#) tab.
2. Search for the entry owner, select it and choose [Add](#).
3. In the [Entry Visibility](#) field, select who should have access to see the role. You can choose between the following:
 - [All](#) - The role is visible to all users. This is the default value.
 - [Owner + members](#) - The role is visible to the owner and the entries that are assigned to the role.
 - [Owner only](#) - The role is only visible to the entries defined as owners.
4. Select another tab to specify other properties of the role or choose [Save](#).

3.4.2.1.5 Specifying Restrictions for a Role

When you configure a role, you use the [Restrictions](#) tab to specify restriction rules for the role.

Context

Here you can specify if the role is allowed or not allowed for selected entries. Additionally, you can use the [Restrictions](#) tab to define "segregation of duty" - roles that are not allowed to be assigned to the same entry at the same time.

For more information about restrictions to role assignments, see [Role and Privilege Attributes](#) in the *SAP Identity Management - Identity Store Schema* document.

To specify restriction rules for the role:

Procedure

1. Select the [Restrictions](#) tab.
2. To add entries that are allowed to have this role, deselect the [Allowed Entries Reversed](#) checkbox, search for an entry, select it and choose [Add](#).

This list contains all entries that are allowed to have this role. If the list contains roles, privileges or (dynamic) groups, this means the members of the given object.

3. To add entries that are **not** allowed to have this role, select the *Allowed Entries Reversed* checkbox, search for an entry, select it and choose *Add*.

This list contains all entries that are **not** allowed to have this role. If the list contains roles, privileges or (dynamic) groups, this means the members of the given object.

4. To add a role to the list of excluded roles, that is, to define roles that are not allowed to be assigned to the same entry at the same time, in the *Exclude Roles* pane, search for the role, select it and choose *Add*.

This list contains other roles which cannot be added to the user simultaneously.

5. Select another tab to specify other properties of the role or choose *Save*.

3.4.2.1.6 Assigning Privileges or Dynamic Groups to a Role

When you configure a role, you use the *Assignments* tabs to specify the privileges or the dynamic groups you want to assign to the role.

Privileges

Context

To assign privileges to the role:

Procedure

1. Select the *Assignments* tab.
2. Search for the privilege to assign to the role. You can do the following:
 - To add a privilege to a role, select it and choose *Add*.
 - To unassign an already assigned privilege to a role, select it and choose *Delete*.
3. Select another tab to specify other properties of the role or choose *Save*.

Dynamic Groups

Context

To assign dynamic groups to the role:

Procedure

1. Select the [Assignments](#) tab.
2. In the [Membership to Role](#) pane, choose [Select](#).
3. In the [Add](#) dialog box, search for the dynamic group you want to assign to the role, select it and choose [Add](#).
4. Select another tab to specify other properties of the role or choose [Save](#).

Note

Have in mind that when assigning a dynamic group to a role, the affected entries are marked as "dirty", and you may need to consider reconciliation and housekeeping. For details, see [Reconciliation \[page 99\]](#) and [Configuring Scheduled Procedures for Housekeeping \[page 116\]](#).

3.4.2.1.7 Configuring Contexts for a Role

When you configure a role, you use the [Context](#) tab to configure context for a role. Normally, only one context is valid for a role.

Context

Note

It is not recommended to change the [Context types](#) once assigned. There is no automatic reconciliation process which will detect this, as the context types are assumed to be unchanged.

To configure context for a role:

Procedure

1. Select the [Context](#) tab.

2. Select an entry type that is defined as context and choose the arrow.
3. Select another tab to specify other properties of the role or choose [Save](#).

Related Information

[Context Based Assignments](#) [page 60]

3.4.2.1.8 Viewing Users Assigned to a Role

When you configure a role, you use the [Assigned to](#) tab to view the users assigned to the role.

Context

To view the users assigned to the role:

Procedure

1. Select the [Assigned to](#) tab.
2. In the [Find](#) field, search for the users assigned to the role.
All entries matching the search criteria are listed.
3. Select another tab to specify other properties of the role or choose [Save](#).

3.4.2.1.9 Building a Role Hierarchy

When you configure a role, you use the [Hierarchy](#) tab to build a role hierarchy.

Context

For more information about role hierarchy and how privileges are inherited in the role hierarchy, see [Role and Privilege Attributes](#) in the *SAP Identity Management - Identity Store Schema* document.

To build a role hierarchy:

Procedure

1. Select the [Hierarchy](#) tab.
2. You can do the following:
 - To add a parent role to the given role, in the [Parent Roles](#) screen area, search for a role, select it and choose [Add](#).
 - To add a child role to the given role, in the [Child Roles](#) screen area, search for a role, select it and choose [Add](#).
3. Select another tab to specify other properties of the role or choose [Save](#).

3.4.2.1.10 Specifying Attester for Role Assignment

When you configure a role, you use the [Attestation](#) tab to specify an attester for role assignment.

Context

The purpose of attestation is to periodically confirm users' access rights to critical resources. Normally, such rights are controlled by assigning specific roles and there is a risk if these roles are assigned to the wrong people. Assignment to these roles should periodically be verified by someone who is responsible for the resource or assignment.

For one given assignment, there is one (and only one) attester.

To specify an attester for the role:

Procedure

1. Select the [Attestation](#) tab.
2. Choose [Select](#).
3. In the [Add](#) dialog box, search for an attester, select it and choose [Add](#).

This will add the attester for the role. To remove the attester, select it and choose [Remove](#).

You can view the status of the attestation, i.e if the attestation is currently active or not. You can also view the date and time when the attestation was initiated last time.

4. Set the date for the next attestation.
5. Select another tab to specify other properties of the role or choose [Save](#).

3.4.2.2 Modifying Roles

Context

To modify a role in the Identity Management User Interface:

Procedure

1. On the *Manage* tab, select *Role* from the *Show* list.
2. Search for the role you want to modify and choose *Choose Task*.
3. In the *Choose Task* dialog box, select ► *Role* ► *Change Role* ► *Choose Task* ►.
4. Select a tab and change the information about the role.

Related Information

[Viewing Attribute History \[page 551\]](#)

3.4.2.3 Deleting Roles

Context

To delete a role in the Identity Management User Interface:

Procedure

1. On the *Manage* tab, select *Role* from the *Show* list.
2. Search for the role you want to delete and choose *Choose Task*.
3. In the *Choose Task* dialog box, select ► *Role* ► *Delete Role* ► *Choose Task* ►.
4. Choose *Delete*.

Note

When a role is removed from a user and an indirectly assigned privilege is removed along with the role, all other roles assigned to this user must be checked to see whether they reference this privilege. Only privileges that are not assigned through other roles can be removed.

3.4.3 Maintaining Privileges

Privileges, also called technical roles, are used to manage provisioning based on assignments of authorizations in target systems. These may be user accounts, e-mail accounts or access control systems.

The fact that a user is given a privilege does not in itself mean that the actual account is created, only that the process has been initiated. For more information, see [Privileges \[page 97\]](#).

You can use the [Tasks](#) and [Member Events](#) tabs to specify processes that are executed when the privilege is assigned to or removed from an entry. The difference is that the processes specified on the [Tasks](#) tab are executed immediately after the privilege is assigned or removed, while the processes specified on the [Member Events](#) tab initiates a pending object that can be used for an approval or any other process to be performed before the privilege is assigned or removed. Either some or all of these processes can be used.

If both the Add member process and the Provisioning process are specified, then the Add member process will be executed first. The Provisioning process will not be started until the Add member process has completed and the privilege actually is assigned. The same will also be the case for the deprovisioning process and the Remove member process. The deprovisioning process will not be executed until the Remove member process has completed.

Privileges can be assigned to a user in the following ways:

- Directly, by adding the privilege to the user. Assigning direct privileges is done by adding the user as a member of the privilege entry (using the MXREF_MX_PRIVILEGE attribute).
- Indirectly, through a role. When assigning a role to the user (by using the MXREF_MX_ROLE attribute of the user entry), all privileges associated with the role (and all sub-roles) will be added to the user. Indirectly assigned privileges are stored in the attribute MX_AUTOPRIVILEGE. The MX_AUTOPRIVILEGE attribute should never be changed by any applications or the Identity Management User Interface, as it is maintained automatically by the Identity Management.

Note

When a role is removed from a user and an indirectly assigned privilege is removed along with the role, all other roles assigned to this user must be checked to see whether they reference this privilege. Only privileges that are not assigned through other roles can be removed.

Note

If you remove an account privilege from a user, only privileges that are directly assigned to this user will be deprovisioned from the respective backend system. Privileges that are assigned through roles need to be cleaned up, for example, by using Validate Remove process on the account privilege.

You can display, search for and maintain privileges in the Identity Management User Interface by using the default forms delivered with the `com.sap.idm.forms.default` package.

By default, identity store entries of all entry reference attributes (for example, all attributes starting with MXMEMBER and MXREF) are sorted out alphabetically by their DISPLAYNAME in Identity Management User Interface (UI). This applies for displaying entries (such as roles, privileges or dynamic groups) in the [Details](#) pane of the UI or viewing the assigned roles, privileges or dynamic groups of a user.

Related Information

[Creating Privileges \[page 523\]](#)

[Modifying Privileges \[page 539\]](#)

[Deleting Privileges \[page 539\]](#)

3.4.3.1 Creating Privileges

Context

To create a privilege in the Identity Management User Interface:

Procedure

1. On the [Manage](#) tab, select [Privilege](#) from the [Show](#) list and choose [Create](#).
2. In the [Choose Task](#) dialog box, choose ► [Privilege](#) ► [Create Privilege](#) ► [Choose Task](#) ►
3. Select a tab to enter information about the privilege.

Related Information

[Specifying the General Properties for a Privilege \[page 524\]](#)

[Specifying Member Events for a Privilege \[page 525\]](#)

[Specifying Approvers for a Privilege \[page 532\]](#)

[Configuring the Event Processes for a Privilege \[page 532\]](#)

[Configuring the Master Privilege for a Privilege \[page 534\]](#)

[Specifying the Visibility of a Privilege \[page 535\]](#)

[Configuring Contexts for a Privilege \[page 536\]](#)

[Viewing Users Assigned to a Privilege \[page 537\]](#)

[Assigning a Privilege to a Role \[page 538\]](#)

3.4.3.1.1 Specifying the General Properties for a Privilege

When you configure a privilege, you use the [General](#) tab to specify the general properties of the privilege.

Context

To specify the general properties of the privilege:

Procedure

1. Select the [General](#) tab.
2. Fill in the required fields. The fields contain the following information:

Field	Description
Unique ID	Enter a name for the privilege. This name must be unique within the identity store (MSKEYVALUE). The recommended syntax is: PRIV:<application name> .
Display Name	Enter a display name for the privilege that is to be used in the Identity Management User Interface.
Repository Name	Select to which repository the privilege is connected. There can also be defined specific repositories for each member event process.
Description	Enter a text describing the privilege.

3. Select another tab to specify other properties of the privilege or choose [Save](#).

3.4.3.1.2 Specifying Member Events for a Privilege

When you configure a privilege, you use the [Member Events](#) tabs to specify member event handling for the privilege.

Context

For more information about processes that are executed when a privilege is assigned to or removed from an entry, see [Member Event Handling \[page 81\]](#).

To specify the member event handling for the privilege, proceed as follows:

Procedure

1. Select the [Member Events](#) tab.
2. Provide the necessary information in the following fields:

Field	Description
Repository	Select the repository that should be used for the Validate Add Process and Validate Remove Process if this is another repository than is defined as default for the privilege. The processes themselves can be defined on the privilege, but it is recommended to define them on the repository type of the referenced repository if the same process can be used for all privileges referencing this repository.
Validate Add Process	Select the process that should be used to verify that the add operation is allowed for the given entry. You can select a process of type Regular or Validate Add Process. This can for example be an approval or a call to GRC Access Control. This process uses the repository specified for the Validate processes. If no repository is specified, the default repository will be used.

Note

Although it is possible to specify the process here, it is recommended that the process is defined on the repository type of the referenced repository as the same process can be used for several roles/privileges referencing this process. In this case, the process here

Field	Description
	is inherited from the repository type of the referenced repository. When Validate Remove process containing an approval task is defined on a child role with assigned privileges, be aware of the following behavior: a removal of the parent role from the user will result in de-assigning of the privileges even though you decline the Validate Removal process on the child role. This process returns either true or false. If this is a process containing an approval task, this is handled automatically, but if you are using a regular process, for instance to check a group membership in an external system, this must be done manually by setting MX_ATTR_STATE to 1 to allow the assignment and 3 to reject it. If this process is defined, the <i>Add Member Process</i> will not be performed unless this process completes successfully. Select a value in the <i>When to Execute Validate Task</i> drop-down list to specify when the process should be executed relative to the valid from and valid to specified for the assignment. You can choose between: • <i>When Valid</i> - Execute the process at the "Valid from" specified in the assignment request. • <i>Immediately</i> - Execute the process when the assignment is requested. This is the default value for validate processes. Note This option works only for direct assignments of privileges. It does not work for privileges assigned through roles. Assignments of roles containing privileges are made as soon as the <i>valid from</i> date is reached, then the role and the privilege will be assigned. • <i>Never</i> - The process will never be executed. • <i>Number of days</i> - Select a number of days before the "Valid from" when the process should be executed. If you select <i>5 days before</i>, the process will be executed 5 days before the date specified as "Valid from" in the assignment request. • <i>Number of weeks</i> - Select a number of weeks before the "Valid from" when the process should be executed. If you select <i>5 weeks before</i>, the process will be

Field	Description
	executed 5 weeks before the date specified as "Valid from" in the assignment request.
Validate Remove Process	Select the process that should be used to verify that the remove operation is allowed for the given entry. You can select a process of type Regular or Validate Remove Process. This can for example be an approval or a call to GRC Access Control. Note If a privilege assignment has expired, it is not possible to approve it. That is, you cannot select a process containing an approval task in the Validate Remove Process field. This process uses the repository specified for the Validate processes. If no repository is specified, the default repository will be used. Note Although it is possible to specify the process here, it is recommended that the process is defined on the repository type of the referenced repository as the same process can be used for several roles/privileges referencing this process. In this case, the process here is inherited from the repository type of the referenced repository. This process returns either true or false. If this is a process containing an approval task, this is handled automatically, but if you are using a regular process, for instance to check a group membership in an external system, this must be done manually by setting MX_ATTR_STATE to 1 to allow removing the assignment and 3 to reject it. If this process is defined, the Remove Member Process will not be performed unless this process completes successfully.
Validate Modify Validity Process	Select the process that should be used to verify that the modify validity operation is allowed for the given entry. You can select a process of type Regular or Validate Modify Process. This can for example be an approval or a call to GRC Access Control. Note Although it is possible to specify the process here, it is recommended that the process is defined on the

Field	Description
	repository type of the referenced repository as the same process can be used for several roles/privileges referencing this process. In this case, the process here is inherited from the repository type of the referenced repository. This process returns either true or false. If this is a process containing an approval task, this is handled automatically, but if you are using a regular process, for instance to check a group membership in an external system, this must be done manually by setting MX_ATTR_STATE to 1 to allow modifying the validity of the assignment and 3 to reject it. If this process is defined, the Modify Member Process will not be performed unless this process completes successfully.
Add Member Process	Select a process that will be performed on the pending value object created when the role is assigned to an entry. This process will perform its operations, for example request an approval when assigning a role. Depending on the status of the operations, the role will or will not be assigned to the entry. The role is not assigned until the process has completed. You can select a process of type Regular or Add Member Process. Any approvers to the Add member process can be added on the Approval tab of an approval task. This process uses the repository specified above. If no repository is specified, the default repository will be used. Note Although it is possible to specify the process here, it is recommended that the process is defined on the repository type of the referenced repository as the same process can be used for several roles/privileges referencing this process. In this case, the process here is inherited from the repository type of the referenced repository. Note If a Validate Add Process is defined, the Add Member Process will not be executed unless the Validate Add Process has completed successfully.

Field	Description
	Select a value in the <i>When to Execute Add Process</i> drop-down list to specify when the process should be executed relative to the valid from and valid to specified for the assignment. You can choose between: • <i>When Valid</i> - Execute the process at the "Valid from" specified in the assignment request. • <i>Immediately</i> - Execute the process when the assignment is requested. This is the default value for validate processes. Note This option works only for direct assignments of privileges. It does not work for privileges assigned through roles. Assignments of roles containing privileges are made as soon as the <i>valid from</i> date is reached, then the role and the privilege will be assigned. • <i>Never</i> - The process will never be executed. • <i>Number of days</i> - Select a number of days before the "Valid from" when the process should be executed. If you select <i>5 days before</i>, the process will be executed 5 days before the date specified as "Valid from" in the assignment request. • <i>Number of weeks</i> - Select a number of weeks before the "Valid from" when the process should be executed. If you select <i>5 weeks before</i>, the process will be executed 5 weeks before the date specified as "Valid from" in the assignment request.
<i>Modify Validity Process</i>	Select a process that will be performed on the pending value object created when the validity of an assignment is modified, either extended or shortened. This process will perform its operations, for example updating the validity in the target application. Depending on the status of the operations, the validity will or will not be modified. The validity is not modified until the process has completed. You can select a process of type <i>Regular</i> or <i>Modify Member Process</i>. This process uses the repository specified above. If no repository is specified, the default repository will be used.

Field	Description
	 Note Although it is possible to specify the process here, it is recommended that the process is defined on the repository type of the referenced repository as the same process can be used for several roles/privileges referencing this process. In this case, the process here is inherited from the repository type of the referenced repository.  Note If a <i>Validate Modify Process</i> is defined, this <i>Modify Member Process</i> will not be executed unless the <i>Validate Modify Process</i> has completed successfully.
<i>Remove Member Process</i>	If the <i>Remove Member Process</i> is specified, the role or privilege will not be removed immediately, but a pending value object is created. Select the process to be executed on this object. The process will then for instance revoke the authorizations in the target system before a privilege is physically removed for the object. The role or privilege is not removed until the process has completed. You can select a process of type <i>Regular</i> or <i>Remove Member Process</i>. This process uses the repository specified above. If no repository is specified, the default repository will be used.  Note Although it is possible to specify the process here, it is recommended that the process is defined on the repository type of the referenced repository as the same process can be used for several roles/privileges referencing this process. In this case, the process here is inherited from the repository type of the referenced repository. If a <i>Validate Remove Process</i> is defined, this <i>Remove Member Process</i> will not be executed unless the <i>Validate Remove Process</i> has completed successfully.
<i>Notification Process</i>	Select a process that will be used to send a notification that an assignment is about to expire. This process is normally specified to be executed a number of days before the assignment expires. You can select a process of type <i>Regular</i> or <i>Notify Process</i>.

Field	Description
	 Note Although it is possible to specify the process here, it is recommended that the process is defined on the repository type of the referenced repository as the same process can be used for several roles/privileges referencing this process. In this case, the process here is inherited from the repository type of the referenced repository. This process may perform actions like: • Send an e-mail to the user that the assignment is about to expire • Extend the assignment by a given period of time • Create an approval to the manager Select a value in the <i>When to Execute Notification Task</i> drop-down list to specify when the process should be executed relative to the valid from and valid to specified for the assignment. You can choose between: • <i>When Valid</i> - Execute the process at the "Valid from" specified in the assignment request. • <i>Immediately</i> - Execute the process when the assignment is requested. This is the default value for validate processes.  Note This option works only for direct assignments of privileges. It does not work for privileges assigned through roles. Assignments of roles containing privileges are made as soon as the <i>valid from</i> date is reached, then the role and the privilege will be assigned. • <i>Never</i> - The process will never be executed. • <i>Number of days</i> - Select a number of days before the "Valid from" when the process should be executed. If you select <i>5 days before</i>, the process will be executed 5 days before the date specified as "Valid from" in the assignment request. • <i>Number of weeks</i> - Select a number of weeks before the "Valid from" when the process should be executed. If you select <i>5 weeks before</i>, the process will be executed 5 weeks before the date specified as "Valid from" in the assignment request.

3. Select another tab to specify other properties of the privilege or choose [Save](#).

3.4.3.1.3 Specifying Approvers for a Privilege

When you configure a privilege, you use the [Approvers](#) tabs to specify approvers for the member event handling.

Context

To specify an approver:

Procedure

1. Select the [Approvers](#) tab.
2. Search for an approver.
3. You can do the following:
 - To add an approver, select the approver and choose [Add](#).
 - To delete an approver, select the already assigned approver and choose [Delete](#).

Note

The default number of escalation levels possible after an approval has timed out is three. You can specify escalation approvers for defined for each escalation level.

4. Select another tab to specify other properties of the privilege or choose [Save](#).

3.4.3.1.4 Configuring the Event Processes for a Privilege

When you configure a privilege, you use the [Tasks](#) tab to specify event processes.

Context

To configure event processes for a privilege:

Procedure

1. Select the [Tasks](#) tab.
2. Provide the necessary information in the following fields:

Field	Description
<i>Provisioning Task</i>	Select the process that is to be executed when the privilege is assigned to an entry. If an Add member process is specified, the Provisioning process will not be executed until the Add member Process has completed. Note Although it is possible to specify the process here, it is recommended that the process is defined on the repository type of the referenced repository as the same process can be used for several roles/privileges referencing this process. In this case, the process here is inherited from the repository type of the referenced repository.
<i>Deprovisioning Task</i>	Select the process that is to be executed when the privilege is removed from an entry. This will normally be the opposite process to the Provisioning process. If a Remove member process is specified, the Deprovisioning process will not be executed until the Remove process has completed. Note Although it is possible to specify the process here, it is recommended that the process is defined on the repository type of the referenced repository as the same process can be used for several roles/privileges referencing this process. In this case, the process here is inherited from the repository type of the referenced repository.
<i>Modify Task</i>	Select the process that is to be executed when any of the attributes are modified. By default, the Modify process is executed when any of the attributes are modified. You can also specify which attributes should trigger the Modify process in the <i>Modify Task Attributes</i> screen area. Note The Modify process will be executed at least once, but may be executed several times if more than one attribute is changed for a user. The same information could then be updated several times in the target systems. It is not possible to prevent multiple updates, so this must be handled in the target systems if it constitutes a problem.

Field	Description
	 Note Although it is possible to specify the process here, it is recommended that the process is defined on the repository type of the referenced repository as the same process can be used for several roles/privileges referencing this process. In this case, the process here is inherited from the repository type of the referenced repository.

3. To specify which attributes should trigger the Modify Process, choose [Add](#), enter a new value and choose [Add](#).
4. Select another tab to specify other properties of the privilege or choose [Save](#).

3.4.3.1.5 Configuring the Master Privilege for a Privilege

When you configure a privilege, you use the [Master privilege](#) tab to configure the master privilege of the privilege.

Context

This is a privilege on which other privileges for the same repository depend. This can be a privilege giving a user an account in the system referred to by the repository definition, which must be assigned before any access rights can be granted this account. Sub-privileges are privileges depending on the presence of the master privilege.

If there is a master privilege defined on the repository, any Add or Remove Member Processes connected to the sub-privileges is not executed until the master privilege is assigned to the user.

To configure the master privilege for the privilege:

Procedure

1. Select the [Master Privilege](#) tab.
2. Choose [Select](#) to select the privilege that is defined as a master privilege for the repository or privilege.

Note

The master privilege policies are only relevant if the privilege being assigned has an Add Member Process defined.

3. Select the process that is executed when a privilege is assigned that requires the presence of the master privilege.

This process can be used to assign the master privilege, by assigning the privilege directly or by assigning a role that references the privilege. When the master privilege is assigned, any assignments waiting for the master privilege will also be assigned.

4. Select another tab to specify other properties of the privilege or choose [Save](#).

3.4.3.1.6 Specifying the Visibility of a Privilege

When you configure a privilege, you use the [Visibility](#) tab to specify who should be allowed to see the privilege.

Context

You can also define an owner of the privilege. The visibility constraint is used in the following cases:

- When searching for entries to which to assign the privilege. Only entries which the user has access to are visible in the search result.
- When managing the privilege. The group members are only visible if the user has access to them.
- When assigning a privilege to a entry, only privileges that the user has access to are visible.

When you specify the visibility of a privilege, you add the entries that should be defined as owner entries. This can either be a user (entry), a privilege, a role or a dynamic group. If the owner is a privilege, role or dynamic group, all their members are considered as owner entries.

To add an entry owner:

Procedure

1. Select the [Visibility](#) tab.
2. Search for the entry owner.
3. You can do the following:
 - To add an entry owner, search for it, select it and choose [Add](#).
 - To delete an entry owner, select the already assigned entry owner and choose [Delete](#).
4. Define the [Access Control for Entry](#) and the [Access Control for Members](#). Select who should have access to see the privilege. You can choose the following:
 - [All](#) - The role is visible to all users. This is the default value.
 - [Owner + members](#) - The role is visible to the owner and the entries that are assigned to the role.
 - [Owner only](#) - The role is only visible to the entries defined as owners.
5. Select another tab to specify other properties of the privilege or choose [Save](#).

3.4.3.1.7 Configuring Contexts for a Privilege

When you configure a privilege, you use the [Context](#) tab to specify how contexts are used for the privilege.

Context

📘 Note

It is not recommended to change the Context types and Autoassign context policy/strategy settings once assigned. There is no automatic reconciliation process which will detect this, as the context types are assumed to be unchanged.

To configure context for a privilege:

Procedure

1. Select the [Context](#) tab.
2. Define the Autoassign context policy.

The autoassign context policy decides how to handle default contexts defined for the users. You can choose between:

- [Inherited](#) - The autoassign context policy is inherited from the default repository.
- [None](#) - Do not use the default contexts defined for the user.
- [If missing](#) - Only add default contexts if the assignment does not already have a context of the given type.
- [Always](#) - Always add the automatically assigned context in addition to the any contexts already assigned.

3. Select an entry type that is defined as context and choose the arrow.
4. To add a conditional context to the privilege, search for it, select it and choose [Add](#).

For privileges, it is also possible to define a list of conditional contexts. In this case, the privilege will only be assigned if one of the specified contexts also is part of the assignment. For instance, when assigning a role with a region as context, only those privileges matching the given region will be assigned.

📘 Note

The configuration of a context (MX_CTX_TYPE) on a privilege without specifying a conditional context (MX_CTX_CONDITIONAL) is not recommended. This could cause inconsistencies between SAP Identity Management and the target system.

Configuring only MX_CTX_TYPE on a privilege and assigning it multiple times through different contexts results in creation of a link in the database for each assignment. In that case, if you afterwards remove one of those assignments from SAP Identity Management, the deprovisioning process will delete the assignment from the target system but will not remove all assignments in SAP Identity Management. Only the assignment with the specified context that you have deleted will be removed from the database.

5. Select another tab to specify other properties of the privilege or choose [Save](#).

Related Information

[Context Based Assignments](#) [page 60]

3.4.3.1.8 Viewing Users Assigned to a Privilege

When you configure a privilege, you use the [Assigned to](#) tab to view users assigned to a privilege.

Context

To view users assigned to a privilege:

Procedure

1. Select the [Assigned to](#) tab.
2. Search for the users assigned to the privilege.
3. All entries matching the search criteria are listed.
4. Select another tab to specify other properties of the privilege or choose [Save](#).

3.4.3.1.9 Assigning a Privilege to a Role

When you configure a privilege, you use the [Hierarchy](#) tab to assign a privilege to a role.

Context

To assigning a privilege to a role:

Procedure

1. Select the [Hierarchy](#) tab.
2. Search for the role you want to assign the privilege to, select it and choose [Add](#).
3. Select another tab to specify other properties of the privilege or choose [Save](#).

3.4.3.1.10 Specifying Attester for Privilege Assignment

When you configure a privilege, you use the [Attestation](#) tab to specify an attester for privilege assignment.

Context

To specify an attester for privilege assignment:

Procedure

1. Select the [Attestation](#) tab.
2. Choose [Select](#).
3. In the [Add](#) dialog box, search for an attester, select it and choose [Add](#).

This will add the attester for the privilege assignment. To remove the attester, select it and choose [Remove](#).

You can view the status of the attestation, i.e if the attestation is currently active or not. You can also view the date and time when the attestation was initiated last time.

4. Set the date for the next attestation.
5. Select another tab to specify other properties of the privilege or choose [Save](#).

3.4.3.2 Modifying Privileges

Context

To modify a privilege in the Identity Management User Interface:

Procedure

1. On the *Manage* tab, select *Privilege* from the *Show* list.
2. Search for the privilege you want to modify and choose *Choose Task*.
3. In the *Choose Task* dialog box, select ► *Privilege* ► *Change Privilege* ► *Choose Task* ►.
4. Select a tab and change the information about the privilege.

Related Information

[Viewing Attribute History \[page 551\]](#)

3.4.3.3 Deleting Privileges

Context

You can delete privileges that you have created in your custom scenarios.

Deleting privileges that are used in the SAP Provisioning Framework for access to a given system makes no sense for the following reasons:

- Deleting these privileges causes inconsistencies in the Identity Management system and does not trigger deprovisioning from the target system.
- If you delete an account privilege `PRIV: <Repository>: ONLY`, you delete it in the Identity Management only. The users that have the account privilege assigned to them are not deprovisioned from the target system. You can still provision any changes to the identity data of the users to the target system.
- Any time you run an Initial Load job, the account privilege that you have deleted, is created again and assigned to the users read from the corresponding repository.
- The system privilege `PRIV: SYSTEM: <Repository>` is an internal privilege for the system and should not be deleted. It is not visible in the Identity Management User Interface.

To delete a privilege in the Identity Management User Interface, proceed as follows:

Procedure

1. On the *Manage* tab, select *Privilege* from the *Show* list.
2. Search for the privilege you want to delete and choose *Choose Task*.
3. In the *Choose Task* dialog box, select ► *Privilege* ► *Delete Privilege* ► *Choose Task* ►.
4. Choose *Delete Entry*.

3.4.4 Maintaining Dynamic Groups

Dynamic groups provide a possibility to select entries (persons) based on attribute values, for instance all with location *Waldorf* or all with title *Manager*. This can also be a combination of several attributes.

You can display, search for and maintain dynamic groups in the Identity Management User Interface by using the default forms delivered with the `com.sap.idm.forms.default` package.

The members of a dynamic group are the entries that match the filter (the attribute values) defined for the group. The attributes `MX_TARGET_FILTER` and `MX_TARGET_DEFINITION` are used to define the members of the dynamic group. Membership in a dynamic group is decided by the values at the time of resolving a group.

Note

You cannot resolve a dynamic group automatically. To resolve the group, use the internal function `uIS_ResolveDynamicGroup` in a custom script. Each time you call this function, you add or remove maximum 1000 entries from a dynamic group. If your SQL query returns 10 000 entries, for example, you need to call `uIS_ResolveDynamicGroup` 10 times to add or remove all the entries.

You can use a dynamic group as follows:

- Use it in a job to define a selection of entries to be processed at the same time, for instance given a certain value. In this case, you can use the dynamic group as a source in a job.
- Use it as a constraint for a role (dynamic roles). In this case any entries matching the filter for the dynamic group will be defined as members of the role. It can also define members which are allowed/not allowed for a role.

Note

When using dynamic groups, consider carefully the performance of any SQL statements used to resolve the group members. Extensive use of dynamic groups is not recommended due to the possible impact on performance.

By default, identity store entries of all entry reference attributes (for example, all attributes starting with `MXMEMBER` and `MXREF`) are sorted out alphabetically by their `DISPLAYNAME` in Identity Management User Interface (UI). This applies for displaying entries (such as roles, privileges or dynamic groups) in the *Details* pane of the UI or viewing the assigned roles, privileges or dynamic groups of a user.

Related Information

[Creating Dynamic Groups \[page 541\]](#)
[Modifying Dynamic Groups \[page 546\]](#)
[Deleting Dynamic Groups \[page 547\]](#)
[ulS_ResolveDynamicGroup \[page 1186\]](#)

3.4.4.1 Creating Dynamic Groups

Context

To create a dynamic group in the Identity Management User Interface:

Procedure

1. On the *Manage* tab, select *Dynamic group* from the *Show* list and choose *Create*.
2. In the *Choose Task* dialog, select ► *Dynamic Group* ► *Create Dynamic Group* ► *Choose Task* ►.
3. Select a tab and fill in the required information.

Related Information

[Specifying the General Properties for a Dynamic Group \[page 542\]](#)
[Defining a Filter for a Dynamic Group \[page 542\]](#)
[Specifying the Visibility for a Dynamic Group \[page 545\]](#)
[Viewing Users Assigned to a Dynamic Group \[page 546\]](#)

3.4.4.1.1 Specifying the General Properties for a Dynamic Group

When you configure a dynamic group, you use the [General](#) tab to specify the general properties for a dynamic group.

Context

To specify the general properties for a dynamic group:

Procedure

1. Select the [General](#) tab.
2. Fill in the required fields. The fields contain the following information:

Field	Description
Unique ID	Enter a unique ID of the dynamic group.
Display Name	Enter a display name for the dynamic group that is to be used in the Identity Management User Interface.
Description	Enter a text describing the dynamic group.

3. Select another tab to specify other properties for the dynamic group or choose [Save](#).

3.4.4.1.2 Defining a Filter for a Dynamic Group

When you configure a dynamic group, you use the [Target Definition](#) tab to define the filter for the dynamic group.

Context

To define the filter for the dynamic group:

Procedure

1. Select the [Target Definition](#) tab.

2. Choose one of the following modes to define the filter (the attribute values) for the dynamic group:

Mode	Description
<i>Simple</i>	In Simple mode, you define the filter for a dynamic group by selecting an attribute name and entering an attribute value in the corresponding fields. The Simple mode supports AND and NOT operations. To define a filter in Simple mode, you have the following options: • Select a single attribute with a single value For example: If you select the MX_TITLE attribute and enter the value manager, you will create a dynamic group with all managers as members. • Select a number of attributes with different values For example: If you select the MX_TITLE attribute and enter the value manager in one field and then select the MX_LANGUAGE attribute and enter the value en in another field, you will create a dynamic group with all managers who speak English. • Select a single attribute with multiple values For example: If you select the MX_LANGUAGE attribute and enter the value en in one field and then select the same attribute with value de in another field, you will create a dynamic group with all users who speak English and German. • Select the <i>Not Equals</i> checkbox You can select the <i>Not Equals</i> checkbox to specify whether an attribute name equals an attribute value. For example: If you enter the MX_LANGUAGE attribute with value en and you select this checkbox, all users who speak languages other than English will become members of the dynamic group.

Advanced

In Advanced mode, you define the filter for a dynamic group by providing a definition in an LDAP format which is then converted to an SQL query.

Note

In SAP Identity Management 8.0, you have to use LDAP format to define filters for dynamic groups instead of writing your own SQL queries (or using an SQL query builder), as in SAP Identity Management 7.2. The LDAP format ensures homogeneous and standardized way of defining dynamic group filters.

The Advanced mode supports the following operations: AND, OR, NOT, Presence and Substring. You can use this mode to define your custom filter (in LDAP format) which is not possible to be defined by using the input fields in Simple mode.

To define a filter in Advanced mode, in the text field, enter an LDAP definition string depending on the supported operations. For example:

- **LDAP syntax for AND operation:**

```
(&(DISPLAYNAME=Admin)(MSKEYVALUE=Administrator))
```

- **LDAP syntax for OR operation:**

```
(|(DISPLAYNAME=Admin)(MSKEYVALUE=Administrator))
```

Mode	Description
	- LDAP syntax for NOT operation: <pre>(!(DISPLAYNAME=Admin))</pre> - LDAP syntax for Presence operation: <pre>(MX_FIRSTNAME=*) (!(MX_MANAGER=*))</pre> <pre>(& (MX_FIRSTNAME=*) (MX_LASTNAME=<...>)) (& (!(MX_MANAGER=*)) (MX_LASTNAME=<...>))</pre> Note If you use the Presence operation for defining a filter based on MSKEYVALUE, DISPLAYNAME or MX_ENTRYTYPE attribute in combination with another attribute, note that the presence of (MSKEYVALUE=*), (DISPLAYNAME=*) or (MX_ENTRYTYPE=*) is of no use, because in any case these three attributes are mandatory for every entry in SAP Identity Management. For example: <pre>(& (MSKEYVALUE=*) (MX_FIRSTNAME=JOHN))</pre> In this example, if an entry with attribute MX_FIRSTNAME=JOHN exists, this means that the entry has an MSKEYVALUE. Using MSKEYVALUE, DISPLAYNAME or MX_ENTRYTYPE attributes in combination with another attributes will impact the performance of the SQL query execution. - LDAP syntax for Substring operation: <pre>(MX_FIRSTNAME=JOHN*) (MX_LASTNAME=*SKI)</pre>

You can switch from Simple to Advanced mode. In this case, the already defined filter for a dynamic group is converted to an LDAP definition string.

Note

If you change an LDAP definition string in the Advanced mode and then switch to Simple mode, you will lose your changes.

3. Select another tab to specify other properties of the dynamic group or choose [Save](#).

3.4.4.1.3 Specifying the Visibility for a Dynamic Group

When you create a dynamic group, you use the [Visibility](#) tab to specify who should be allowed to see the dynamic group.

Context

When you specify the visibility of a dynamic group, you add the entries that should be defined as owner entries. This can either be a user, a privilege, a role or a dynamic group. If the owner is a privilege, role or dynamic group, all their members are considered as owner entries.

To specify who should be allowed to see the dynamic group:

Procedure

1. Select the [Visibility](#) tab.
2. You can do the following:
 - Search for the entry owner, select it and choose [Add](#).
 - To delete an entry owner, select the already assigned entry owner and choose [Delete](#).
 - Define the Access Control for Entry and the Access Control for Members. You need to select who should have access to see the dynamic group. You can choose between:
 - [All](#) - The dynamic group is visible to all users. This is the default value.
 - [Owner + members](#) - The dynamic group is visible to the owner and the entries that are assigned to the dynamic group.
 - [Owner only](#) - The dynamic group is only visible to the entries defined as owners.
3. Select another tab to specify other properties of the dynamic group or choose [Save](#).

3.4.4.1.4 Viewing Users Assigned to a Dynamic Group

When you create or modify a dynamic group, you use the [Assigned Users](#) tab to search and view the users assigned to the dynamic group.

Context

To search and view the users assigned to a dynamic group:

Procedure

1. Select the [Assigned Users](#) tab.
2. Search for the users assigned to the dynamic group.
3. All entries matching the search criteria are listed.
4. Select another tab to specify other properties of the dynamic group or choose [Save](#).

3.4.4.2 Modifying Dynamic Groups

Context

To modify a dynamic group in the Identity Management User Interface:

Procedure

1. On the [Manage](#) tab, select [Dynamic group](#) from the [Show](#) list.
2. Search for the dynamic group you want to modify and choose [Choose task](#).
3. In the [Choose Task](#) dialog, select ► [Dynamic Group](#) ► [Modify Dynamic Group](#) ► [Choose Task](#) ►.
4. Select a tab and change the information about the dynamic group.

3.4.4.3 Deleting Dynamic Groups

Context

To delete a dynamic group in the Identity Management User Interface:

Procedure

1. On the *Manage* tab, select *Dynamic group* from the *Show* list.
2. Search for the dynamic group you want to delete and choose *Choose task*.
3. In the *Choose Task* dialog, select ► *Dynamic Group* ► *Delete Dynamic Group* ► *Choose Task* ▾.
4. Choose *Delete*.

3.4.5 Handling Basic and Assignment Approvals

Approvers can view detailed information about basic and assignment approvals and handle them in the Identity Management User Interface.

Prerequisites

The logged-in user has access to the *To Do* tab, that is, he or she has the MX_PRIV:WD:TAB_TODO privilege assigned.

Context

Proceed as follows:

Procedure

1. Select the *To Do* tab and search for approvals.
 - In the *Show* dropdown menu, you can filter approvals by operation.

[Add](#), [Modify](#) and [Delete](#) display assignment approvals for the respective operation. [Others](#) displays basic approvals.

- In the [Find](#) field, you can search for approvals by a consignee name, an assignment and a context (if any).

2. Select the approval you want to view the details for.

How many tabs you see in the [Entry Data](#) pane depends on the type of the approval.

Tab	Details
Assignment Information	Available for assignment approvals. Displays information about the type of operation, the consignee, the validity of the assignment, the assignment itself, the context (if a role or a privilege is assigned with a given context) and description.
Request Information	Available for assignment approvals. Displays information about who requested the assignment, for what reason and when it was requested.
Approval Information	Available for assignment and basic approvals. Displays information about the timeout of the approval, the number of required approvers to approve the request, number of approvals performed and the Approvers table.

📘 Note

For **assignment approvals**, the [Approvers](#) table displays the list of all approvers calculated at the time the assignment approval is initialized. If conditions change during the approval process, no approvers will be added or removed from the original list.

For example: If you specify two approvers for a role and assign this role to a user, both approvers will be listed in the [Approvers](#) table for the assignment approval. If one of the approvers is removed from the role during the approval process, he or she will not be removed from the original list in the [Approvers](#) table for this assignment approval.

📘 Note

For **basic approvals**, the [Approvers](#) table displays only those approvers that have already accessed their [To Do](#) tab where approvals for which they are identified as approvers are listed. This means that the number of approvers you see at a given time, might not be the complete number of approvers.

Tab	Details
Approval History	Available for assignment approvals. Displays information about the user who performed some action (for example, the user requesting the assignment or the user delegating an approval), the operation performed (for example, Assignment requested, Delegated to), the user to whom an approval is delegated, date and reason of the event.
Attributes	Available for assignment and basic approvals. This tab is displayed if Show Attributes is selected on the Approval tab of the approval task. Note For basic approvals, the approver can view and edit attributes of the entry being approved. For assignment approvals, the approver can view and edit the pending value object. If one of the approvers delegates an approval, the new approver can access the approval's attributes.

- Handle the approval.

Note

Due to the specifics and the design of the Approval mechanism, we do not support the combination of grouping and approvals and we do not recommend using them together. However, if you decide to combine them, some faulty behavior may be observed. For more information, see [3196414](#) .

Related Information

[Approval Processing \[page 34\]](#)

3.4.6 Viewing Task History

In the Identity Management User Interface, you can view the status and history of self-service tasks, tasks available from the [Manage](#) tab and approvals.

Prerequisites

The logged-in user has access to the [History](#) tab, that is, he or she has the MX_PRIV:WD:TAB_HISTORY privilege assigned.

Context

Proceed as follows:

Procedure

1. Select the [History](#) tab.
2. In the [Show](#) field, select the tasks from the dropdown menu. You have the following options:
 - [Self-Service Tasks](#) - tasks executed on own entry
 - [Management Tasks](#) - tasks executed on other entries
 - [Approvals](#) - approval tasks

Note

The [Reason](#) column that is displayed for each task matches the MSG (messages) column in the mxwv_audit_simple database view. Note that it does not provide the reason for requesting or executing a task (for example, modifying an assignment).

3. You can filter tasks by date or by using the [Find](#) field, where you can only filter tasks by providing the task name.

Note

The [Details](#) pane on the [History](#) tab is no longer available as of version 8.0 SP07 SAP Identity Management User Interface patch level 6 and higher.

3.4.7 Viewing Attribute History

The *Historical Values* tab of a form in the Identity Management User Interface displays information about modified or deleted attributes.

Prerequisites

You have enabled the option *Show Historic Values* of a form in the SAP Identity Management Developer Studio.

Context

Based on the selected type in the *Attributes list* of the form, different attributes can be displayed in the *Historical Values* tab.

To attain the information, proceed as follows:

Procedure

1. Select the *Historical Values* tab of a form.
2. Preview the information or filter it by any attribute, displayed in the tab.

The following attributes may be displayed in the Historic Values tab:

- *Unique ID* - name of the attribute
- *Attribute Name* - *Display name* of the attribute
- *Value* - Based on the operation type performed, this column displays:
 - In case of *Modify* operation - the previous value of the attribute
 - In case of *Delete* operation - the value of the attribute that has been deleted
- *Context* - context related to this attribute
- *Operation* - type of the performed operation
 - *Modify* - when the attribute value is changed
 - *Delete* - when the attribute value is removed
- *Modified* - date and time of the last modification
- *Modified by* - indicates how the last modification for this entry was done.
 - *MSKEYVALUE* - When the attribute value is modified via the Identity Management User Interface, the column displays the MSKEYVALUE of the user who performed the operation.
 - *Runtime engine* - when the attribute value is added or modified via a job
 - *Reconcile* - when the attribute value is added after reconciliation
 - *Dynamic group expansion* - when the attribute is added via dynamic group recalculation
 - *System* - when the attribute value is added or modified via stored procedure

- *Unknown* - in any other cases
- *Removed by* - indicates how the attribute was removed.

ⓘ Note

For operation *Modify* the *Removed by* column remains empty.

- *MSKEYVALUE* - MSKEYVALUE of the user who deleted the attribute via the Identity Management User Interface
- *Dynamic group* - when the attribute is removed due to recalculation of a dynamic group
- *Validity expiration* - when the attribute is removed because of validity expiration. This also applies if the *Time to Live* option is set for an attribute.
- *Reconcile* - when the attribute is removed after reconciliation
- *Runtime engine* - when the attribute is removed via a job

3.4.8 Searching for Identity Store Entries

When searching for identity store entries in the Identity Management User Interface, you can use basic or advanced search. The advanced option allows you to narrow down the search result more than you can by using the basic search.

Context

The attributes that can be used in basic and advanced search depend on the definition of a search and/or display form for the respective entry type.

- If a search form is defined for the entry type, any attribute contained in this form can be used.
- If only a display form is defined for the entry type, any attribute contained in this form can be used.
- If neither a search nor display form is set, any attribute that is selected in the List column of the entry type can be used.
- If also no attribute in the List column is selected, the default search attribute is the MSKEYVALUE.

ⓘ Note

Searching by entry reference attributes is supported in *Advanced* search only.

If you want to search for entries by entry reference attributes, such as MX_MANAGER, you need to provide the MSKEYVALUE of the referenced entry.

Proceed as follows:

Procedure

1. On the [Manage](#) tab, select an entry in the [Show](#) field and search for an entry based on the attribute definition of your search or display form.
 - For basic search, enter your search criteria and choose [Go](#).
 - For advanced search, choose the [Advanced](#) option, enter your search criteria and choose [Go](#).
For example, search for employees by the MSKEYVALUE of their manager.
2. You can switch between basic and advanced search options.

3.5 Self-Service Password Reset

You can configure and implement self-service password reset in SAP Identity Management 8.0 SP00 or higher. In addition to this solution, you can also use the Logon Help to change passwords.

Prerequisites

The following software is required:

- SAP Identity Management 8.0 SP00 or higher, correctly installed and licensed.
- For setting of the productive password in UME one of the following SAP NetWeaver versions is required:
 - SAP NetWeaver 7.3 SP09 or higher
 - Enhancement Package 1 for SAP NetWeaver 7.3 SP06 Patch 3 or higher
 - SAP NetWeaver 7.4 SP02 or higher
- At least one dispatcher has been configured and is running.
- An identity store with at least one user (in addition to admin user).
- An Identity Management User Interface has been configured according to *Installing and Configuring SAP Identity Management User Interface*

Overview

The password reset process consists of the following three steps:

1. **Identify:** In this step, the user will be asked for the unique identifier, the default is MSKEYVALUE. Other options are to ask for another unique attribute (for example, email address) in addition to or instead of the MSKEYVALUE. This is configured on the Password Reset form in the Identity Management Developer Studio.

2. **Verify identity (authenticate):** The user answers some question(s) only he/she knows the answer to. The answers are not available from observation, public records, or social media.

It will be possible to define any number of questions on a system, using the attributes on the format MX_AUTHQ_nnn (for example, MX_AUTHQ_001, MX_AUTHQ_002 etc). Five (5) attributes are defined by default, but any implementation may add additional attributes following the naming syntax. A form needs to be created where the user is required to answer a minimum number of these questions. That is, you can define how many of the defined questions the user has to answer. See *Setting Password Reset Parameters* and *Creating Edit Authentication Questions Form* in the Related Information section. You can also define a maximum number of login attempts in the identity store configuration.

Note

Every failed attempt of password reset is logged, and a task is executed. For security reasons, the user is not told why a password reset attempt failed, if too many attempts are made to reset the password or if the provided unique identifier does not exist, as this would provide a potential attacker with additional information. Instead, the password reset will proceed but random authentication questions may be displayed to the user (including the ones that user has not defined the answers for) and the password reset will fail regardless of the input information being correct or not.

3. **Set password:** A new password is provided to the user, either as input by the user (stored in the attribute MX_PASSWORD) or system generated. In either case, the password is validated towards the UME (User Management Engine) password policy, and a task is started. This task can then perform any desired operations, for example, sending the new password to the user via e-mail or SMS, provision the password and so on.

Note

If the password is not accepted upon the validation towards the UME password policy, then nothing is written to the database (or the UME) and the user may try to set the password again.

The questions used for password reset authentication are system specific. That is, all users in the same identity store will have the same questions available and in the same language (the authentication questions are available in several languages). By default, the following questions are given:

- What is your favorite color?
- What make of car do you drive?
- What is your pet's name?
- What is your mother's maiden name?
- What street did you grow up on?

The questions may be changed by altering the display name of attributes MX_AUTHQ_001 to MX_AUTHQ_005. See *Changing Authentication Questions* in the Related Information section.

Note

This change of questions should be done during the implementation of the self-service password reset, and it is especially important that this is done before the form allowing the users to enter answers to the authentication questions is available for the users. Changing the questions after the users have provided their answers will cause the answers not to fit well any more.

Related Information

[Installing and Configuring SAP Identity Management User Interface](#)

[Creating UME Role `idm.anonymous` \[page 555\]](#)

[Creating Password Reset Form \[page 562\]](#)

[Creating Password Reset Failed Process \[page 565\]](#)

[Configuring Password Reset Parameters \[page 569\]](#)

[Editing Authentication Information \[page 573\]](#)

[Testing Self-Service Password Reset \[page 578\]](#)

[Changing Authentication Questions \[page 583\]](#)

[Resetting Number of Failed Password Reset Attempts \[page 587\]](#)

[Setting Password Reset Parameters \[page 571\]](#)

[Creating Edit Authentication Questions Form \[page 574\]](#)

3.5.1 Creating UME Role `idm.anonymous`

Context

Before configuring the password reset forms and processes, the following needs to be in place:

- At least one user in the identity store (in addition to the admin user).
- A UME role with action `idm_anonymous` assigned to group *Anonymous Users* in the User Management Engine (in addition to roles described in *Installing and Configuring SAP Identity Management User Interface*).

To create the UME role `idm.anonymous`, proceed as follows:

Procedure

1. Log on to SAP NetWeaver Administrator at `http(s)://<host>:<port>/nwa` and navigate to  *Configuration*  *Identity Management* .
2. In *Identity Management: Overview*, change *Search Criteria* to *Role* and then choose *Create Role*.

Identity Management - SAP NetWeaver Administrator - Internet Explorer

http://[redacted]webdynpro/resources/sap.com/tc~lm

Identity Management - SAP ...

Identity Management: Overview

Restore Default View | Back Forward | History | Home | Help | Log Off

Favorites | Related Links | Go To | Support Details

Search: Go

Identity Management | Import | Configuration | Consistency Check

Search

Search Criteria: Role | All Data Sources | Go

Create Role | Delete | Export

Principal Type	Name	Description	Data Source
Enter a search parameter and choose Go			

Details

Modify | Save | Cancel

General Information | Assigned Groups | Assigned Users | Assigned Actions | User Mapping for System Access

Unique Name: *

Display Name:

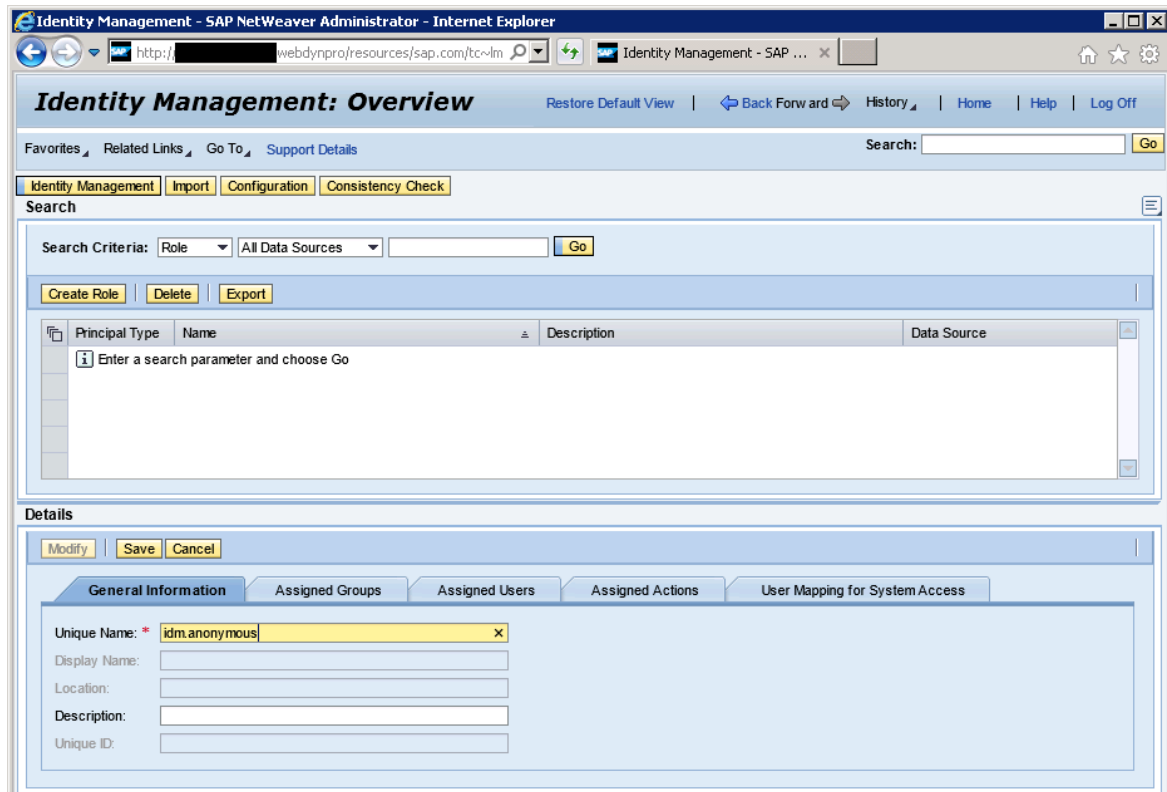
Location:

Description:

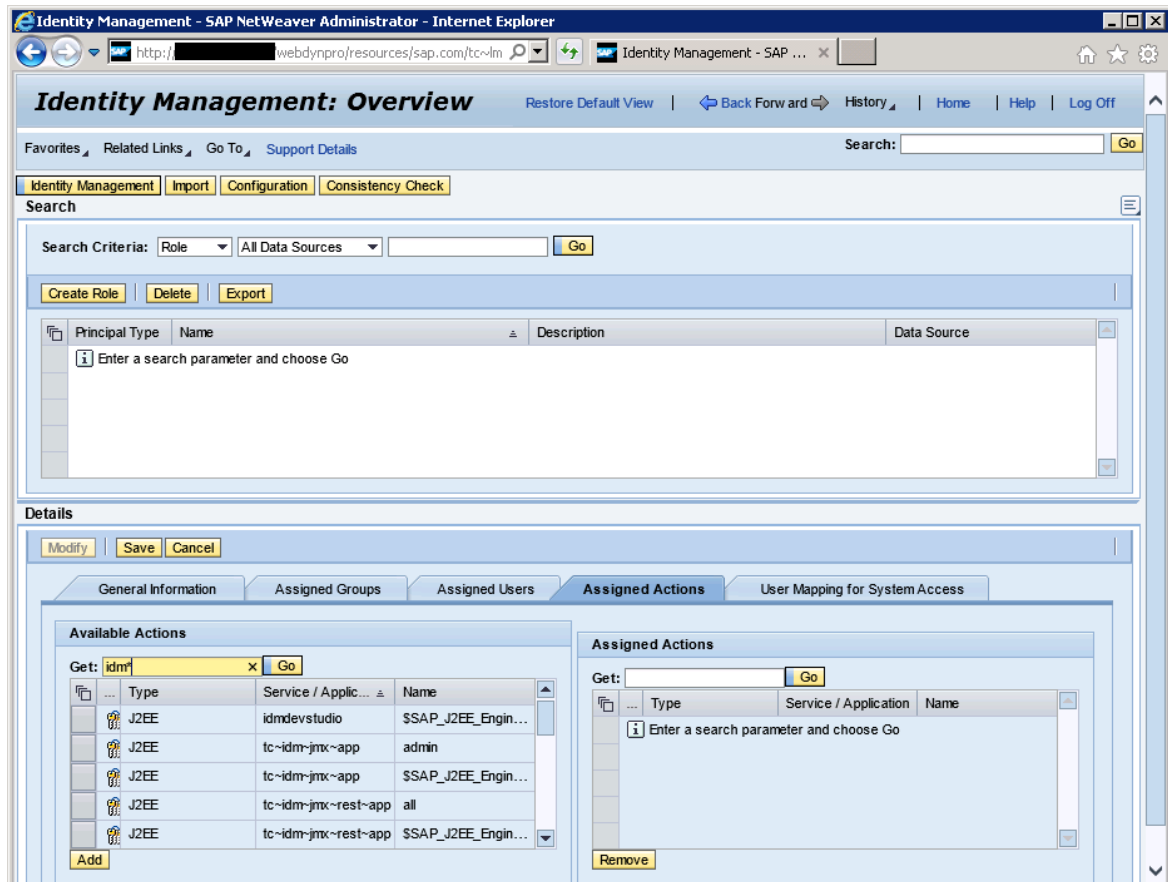
Unique ID:

3. Select the *General Information* tab and fill in the following:

- In the *Unique Name* field, give the role a descriptive name (here **idm.anonymous**).
- In the *Description* field, enter a short description of the role. This is not a mandatory field.

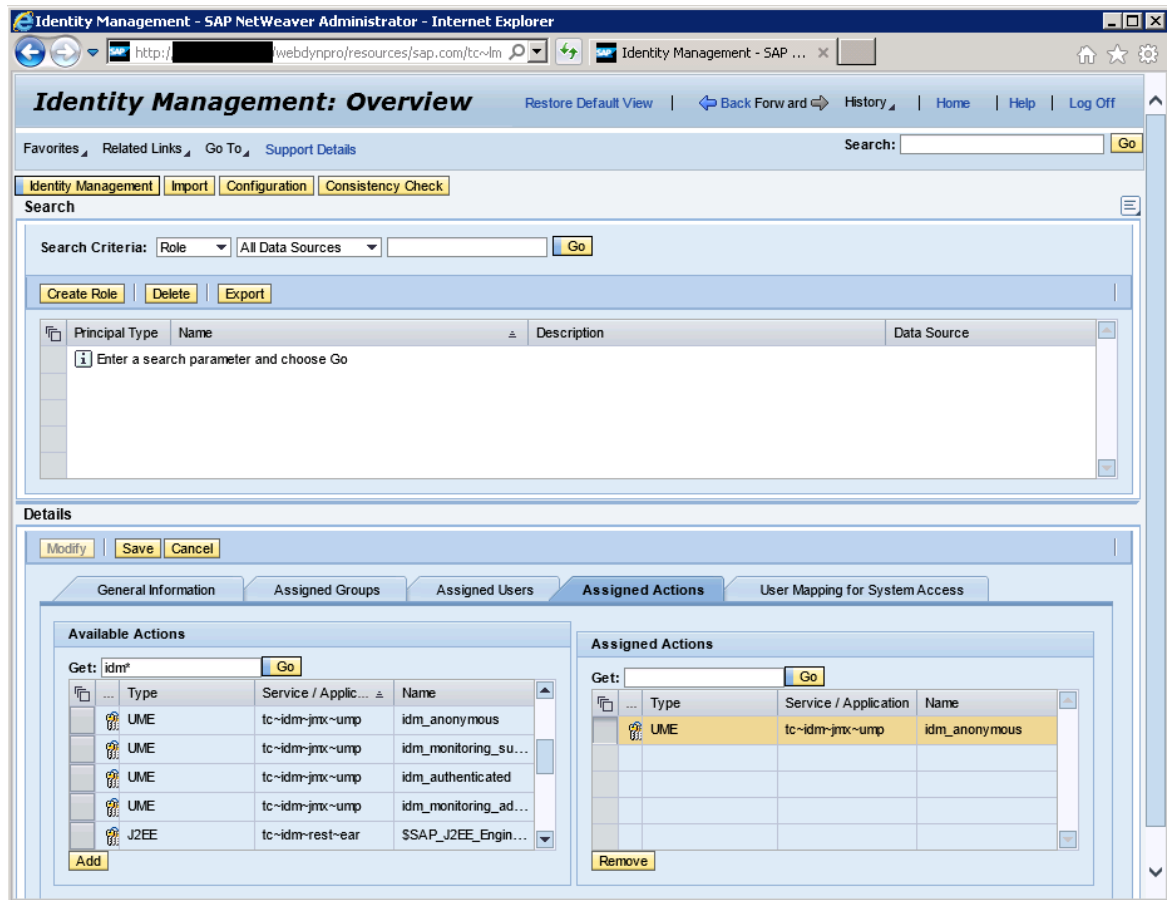


4. Select the *Assigned Actions* tab.
5. In the left pane *Available Actions*, enter **idm*** in the *Get* field and choose *Go*.



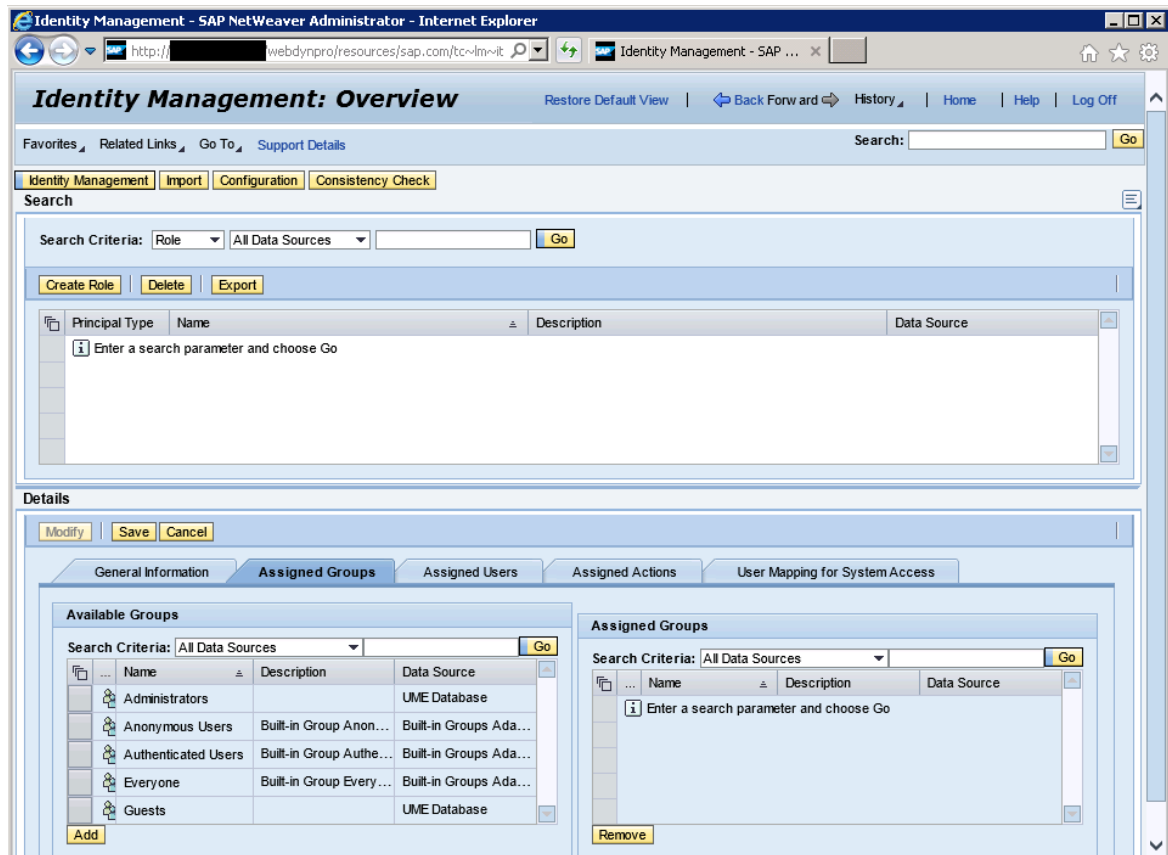
A list of actions/access rights that you can link to the role is displayed.

6. Select the *idm_anonymous* action and choose *Add*.

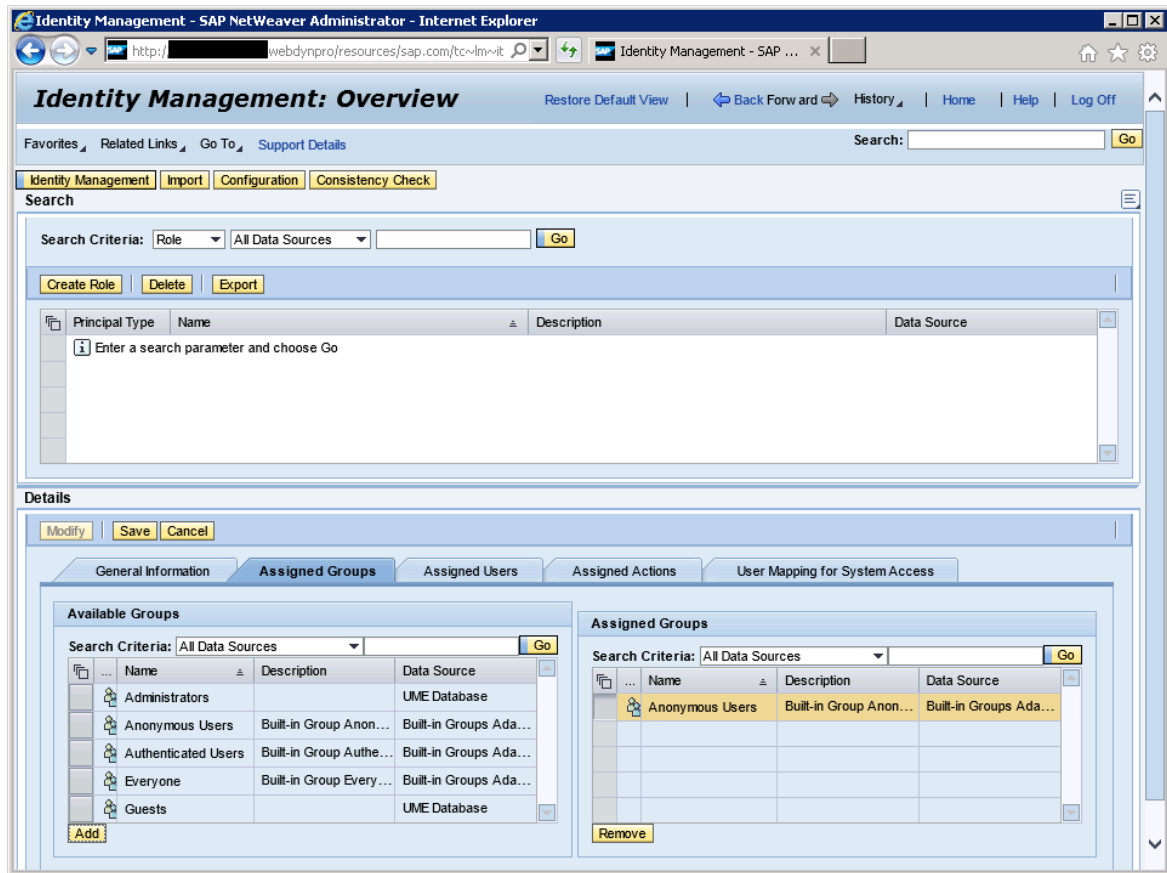


The *idm_anonymous* action is now assigned to the role and this is displayed in the right pane *Assigned Actions*.

7. Select the *Assigned Groups* tab.
8. In the *Available Groups* pane, choose *Go* to list all available groups.

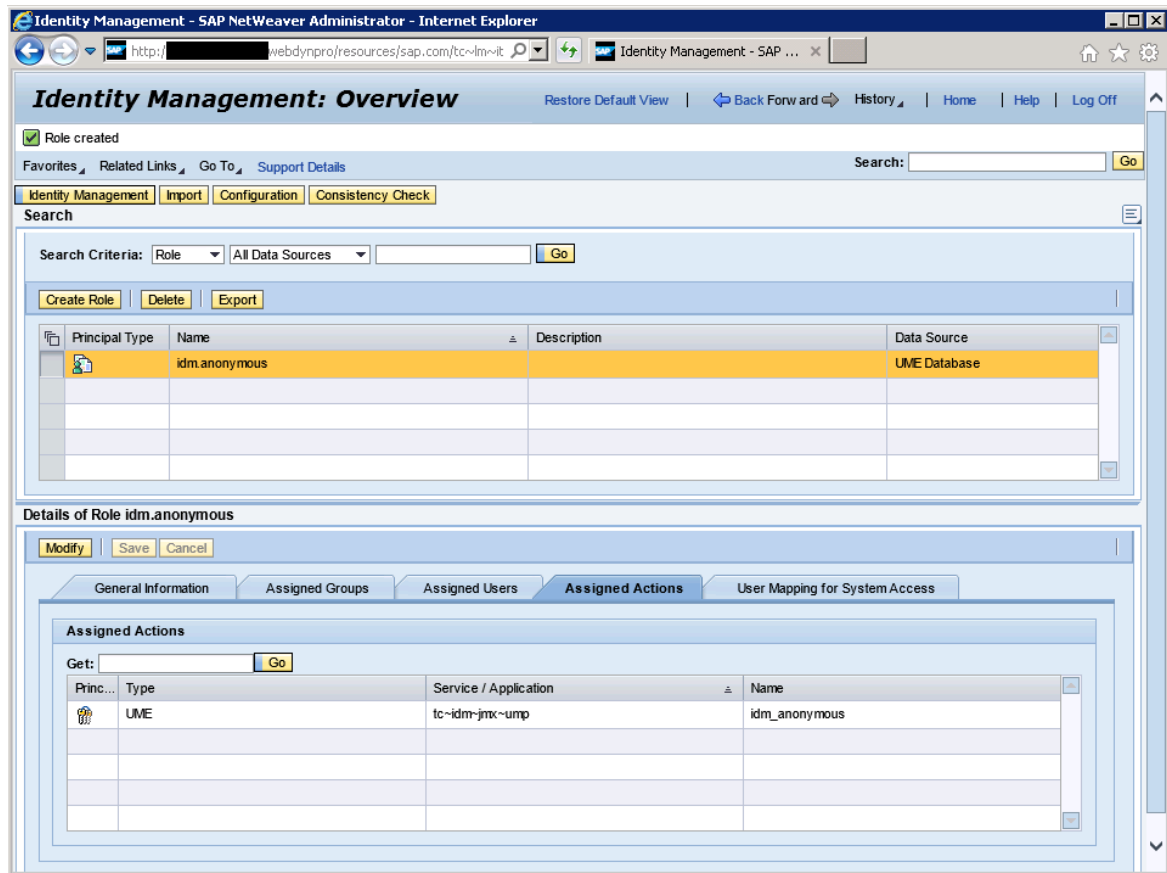


9. Select the *Anonymous Users* group and choose *Add*.



The *Anonymous Users* group is now given the role and this is displayed in the right pane *Assigned Groups*.

10. Choose [Save](#) to confirm and create the new role, which will give access to the password reset to every anonymous user. The just created role will be displayed in the list of the roles available:



Related Information

[Installing and Configuring SAP Identity Management User Interface](#)

3.5.2 Creating Password Reset Form

The password reset form (UI task) is used to generate a new password for a user who has forgotten his/her password. The form is then added to the identity store configuration so that it will be available for the anonymous users.

Prerequisites

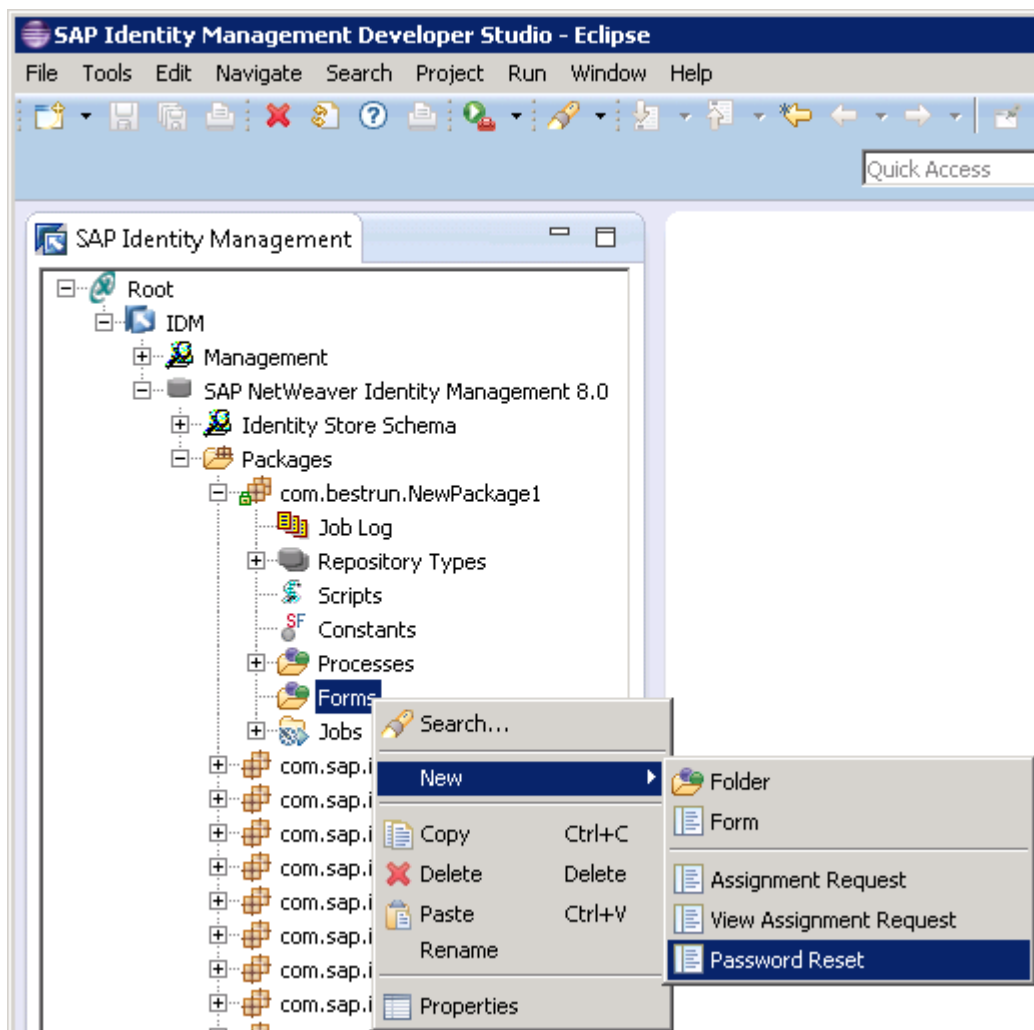
- The package where you want to create the Password Reset form is checked out and ready for editing.

Context

To create the password reset form, proceed as follows:

Procedure

1. In the Identity Management Developer Studio tree view, select the package where you want to create the Password Reset form.
2. Select the *Forms* folder and choose ► *New* ► *Password Reset* .

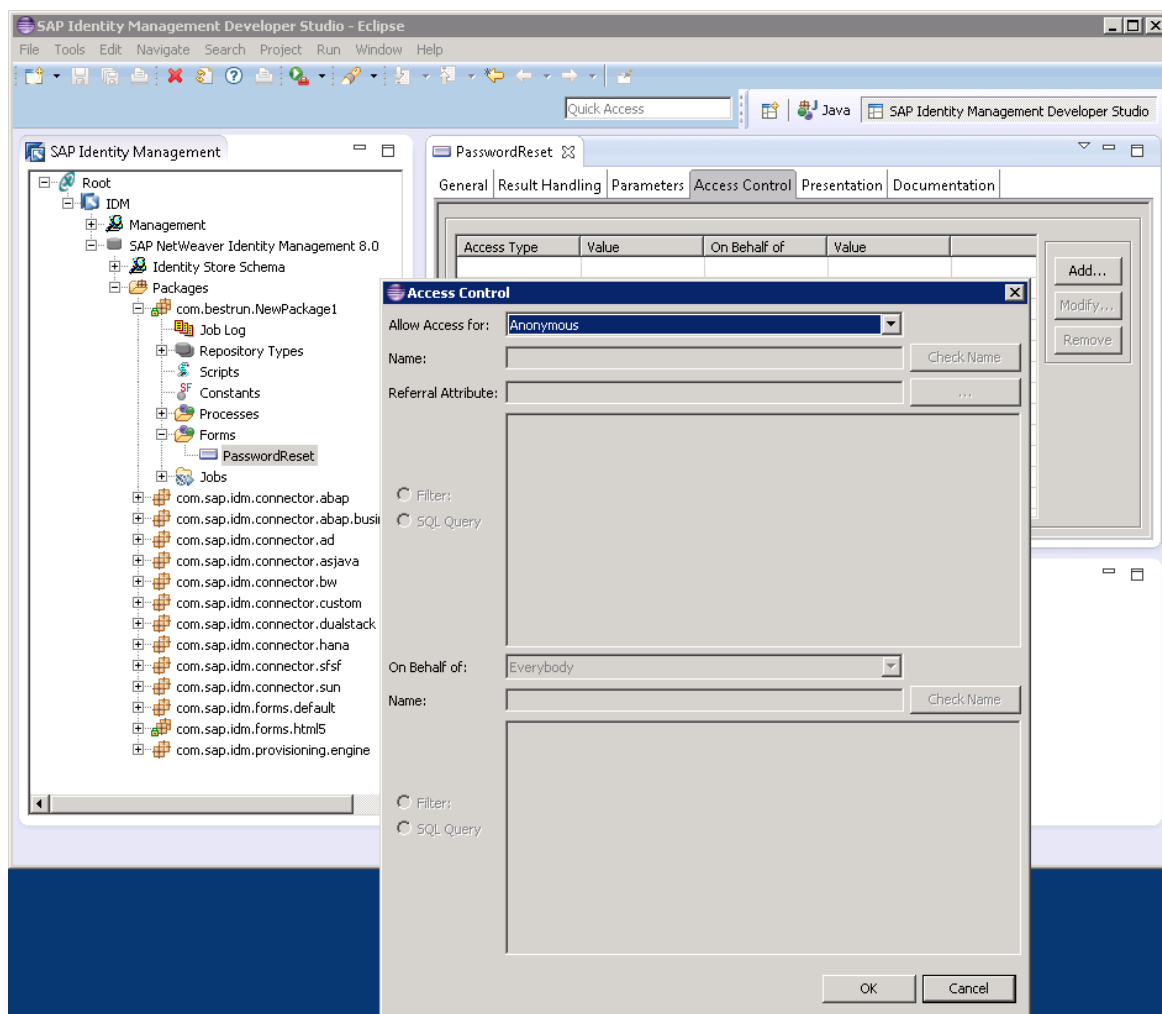


3. To modify the name of the form, choose *Rename* from the context menu and enter **PasswordReset**, for example.

Note

The name can contain only letters (a-z), (A-Z), numbers (1-9) and underscore (_). Spaces are not supported.

4. Select the *Access Control* tab and choose *Add*.
5. In the *Access Control* dialog box, select *Anonymous* in the *Allow Access for* field and choose *OK*.



6. Choose *Save*.

Results

The Password Reset form is now defined.

3.5.3 Creating Password Reset Failed Process

The Password Reset Failed process runs every time the Password Reset form fails. This process can be configured to do several things upon the password reset error. In this document, the process creates an ASCII file and logs the error information.

Prerequisites

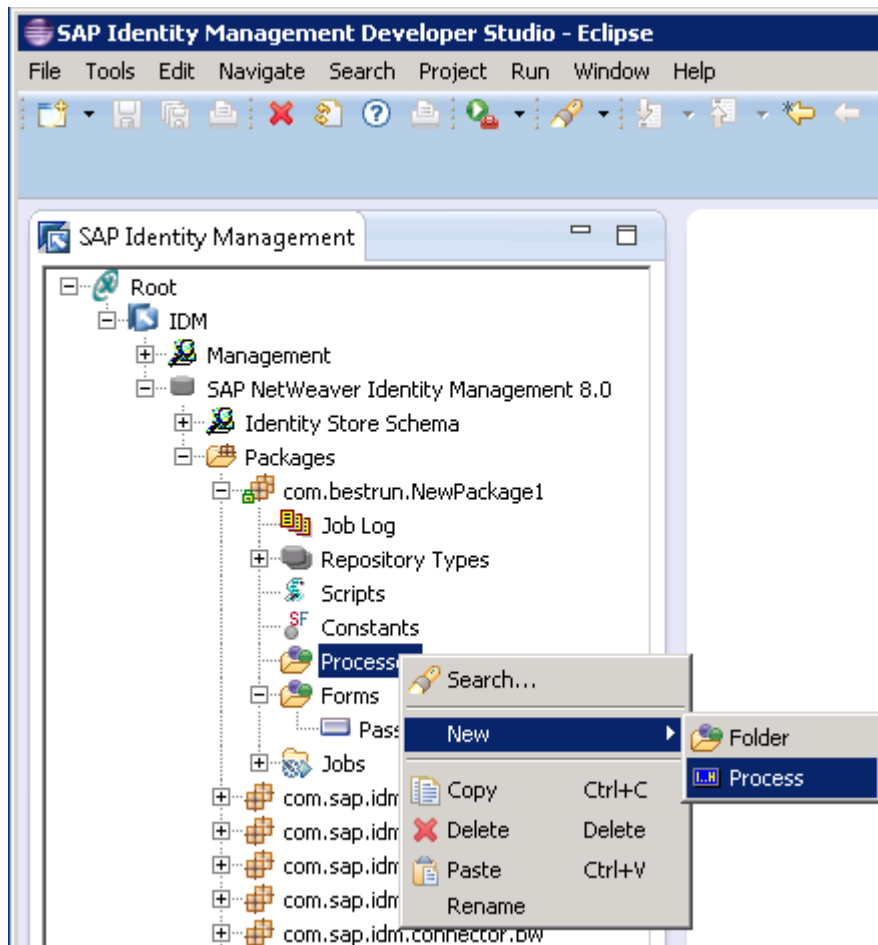
- The package where you want to create the Password Reset Failed process is checked out and ready for editing.

Context

To create the Password Reset Failed process, proceed as follows:

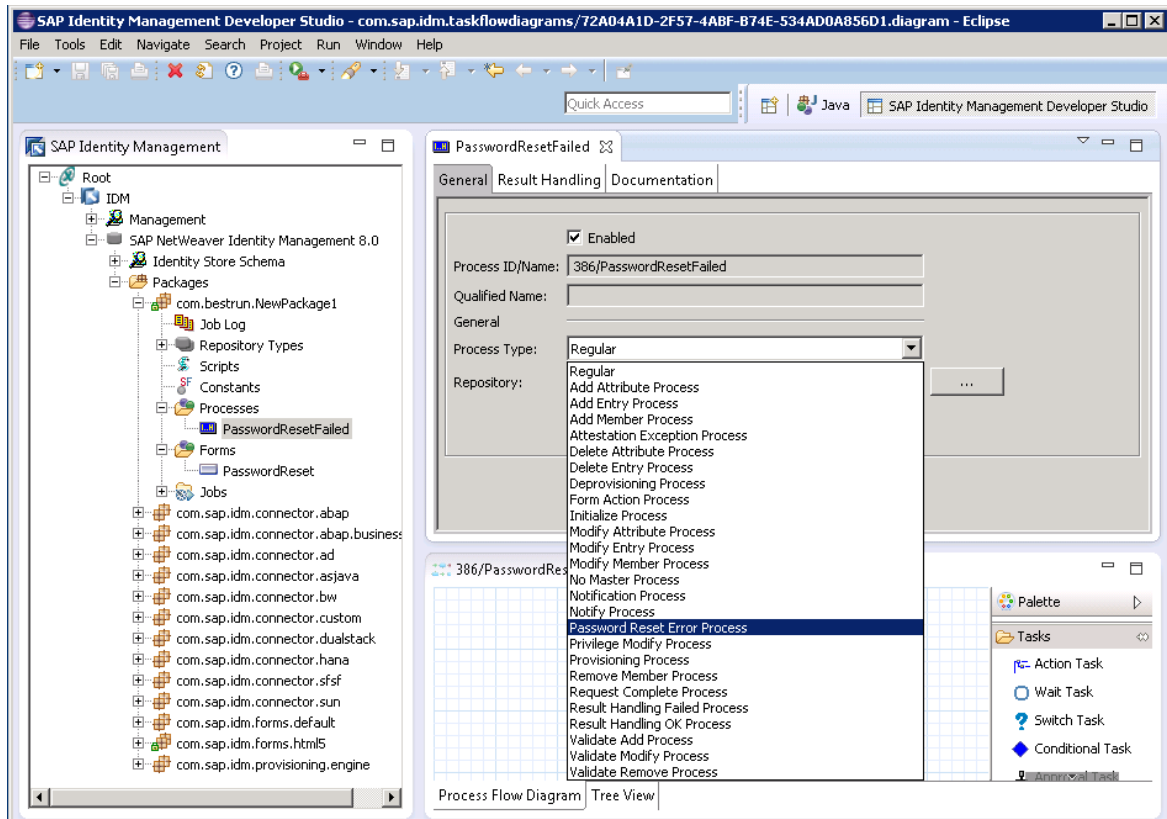
Procedure

1. In the Identity Management Developer Studio tree view, select the package where you want to create the Password Reset Failed process.
2. Select the *Processes* folder and choose  *New*  *Process*  from the context menu.

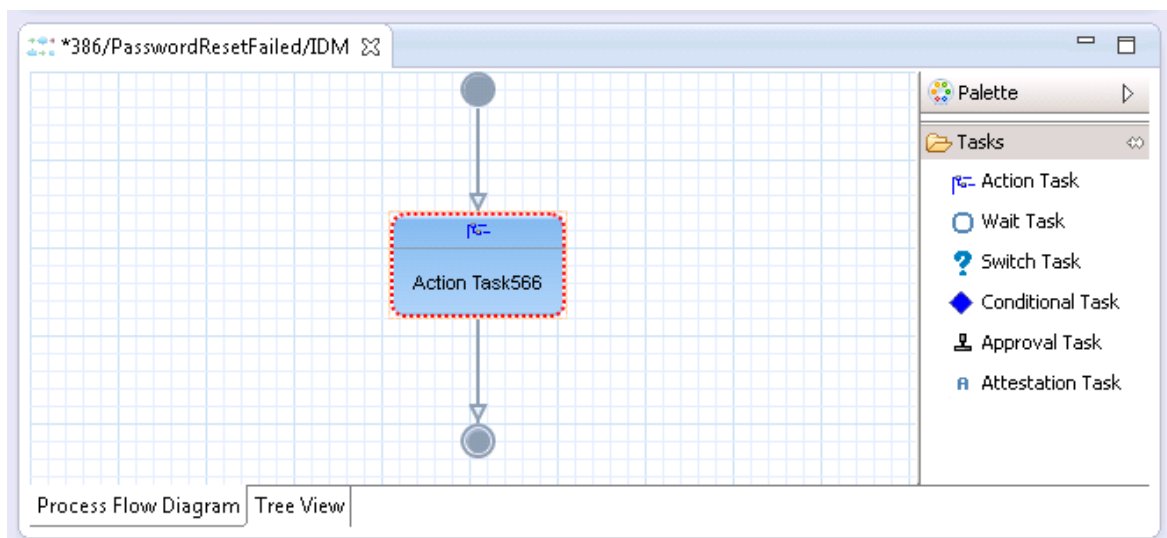


The process flow diagram opens automatically when the process is created.

3. To modify the name of the process, choose *Rename* from the context menu and enter **PasswordResetFailed**, for example.
4. On the *General* tab of the *PasswordResetFailed* process, define the type of the process to be *Password Reset Error Process*.



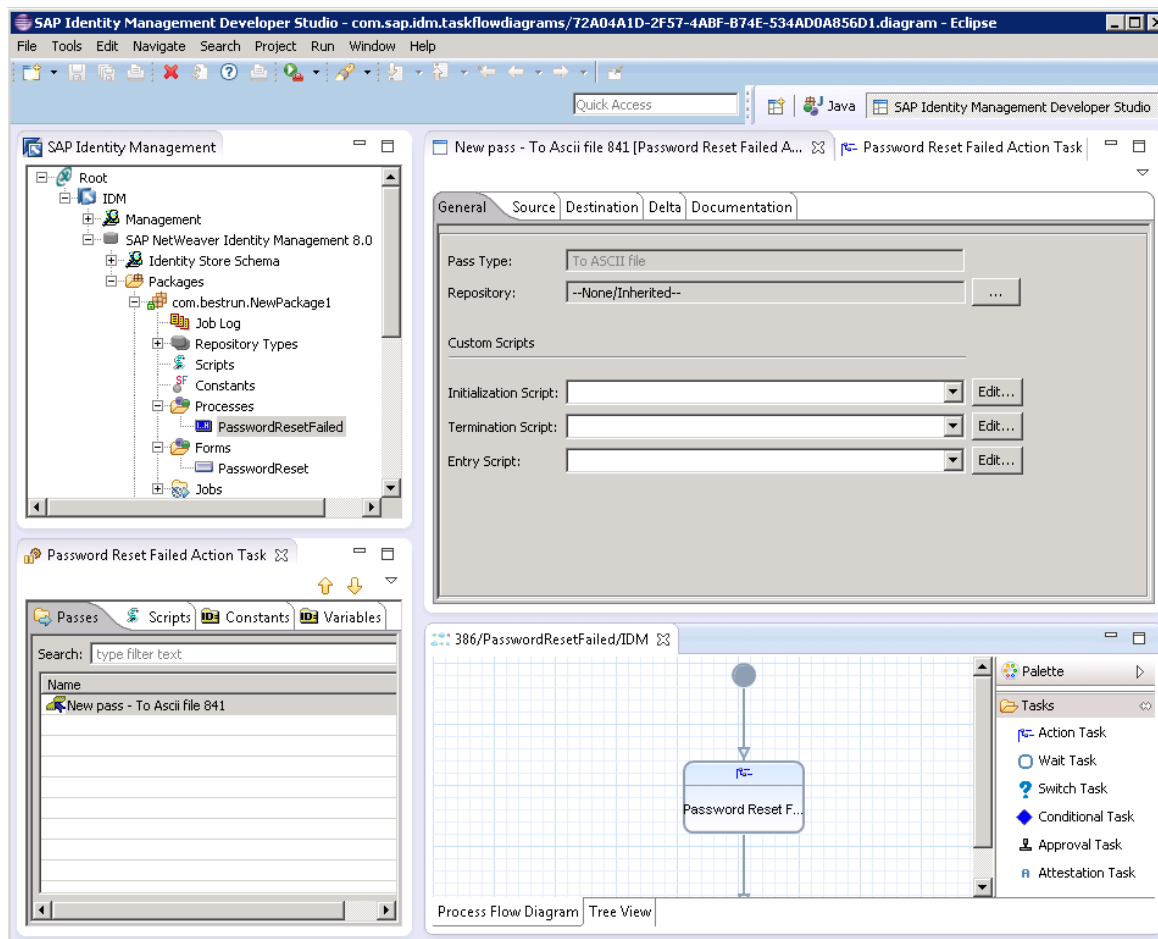
5. Choose **Save**.
6. Use the process flow diagram to graphically model the process and its content.
7. Select **Action Task** from the palette to the right of the canvas and move the cursor to the connection where you want to add the task. When the connection is highlighted (yellow and dotted), drop the task.



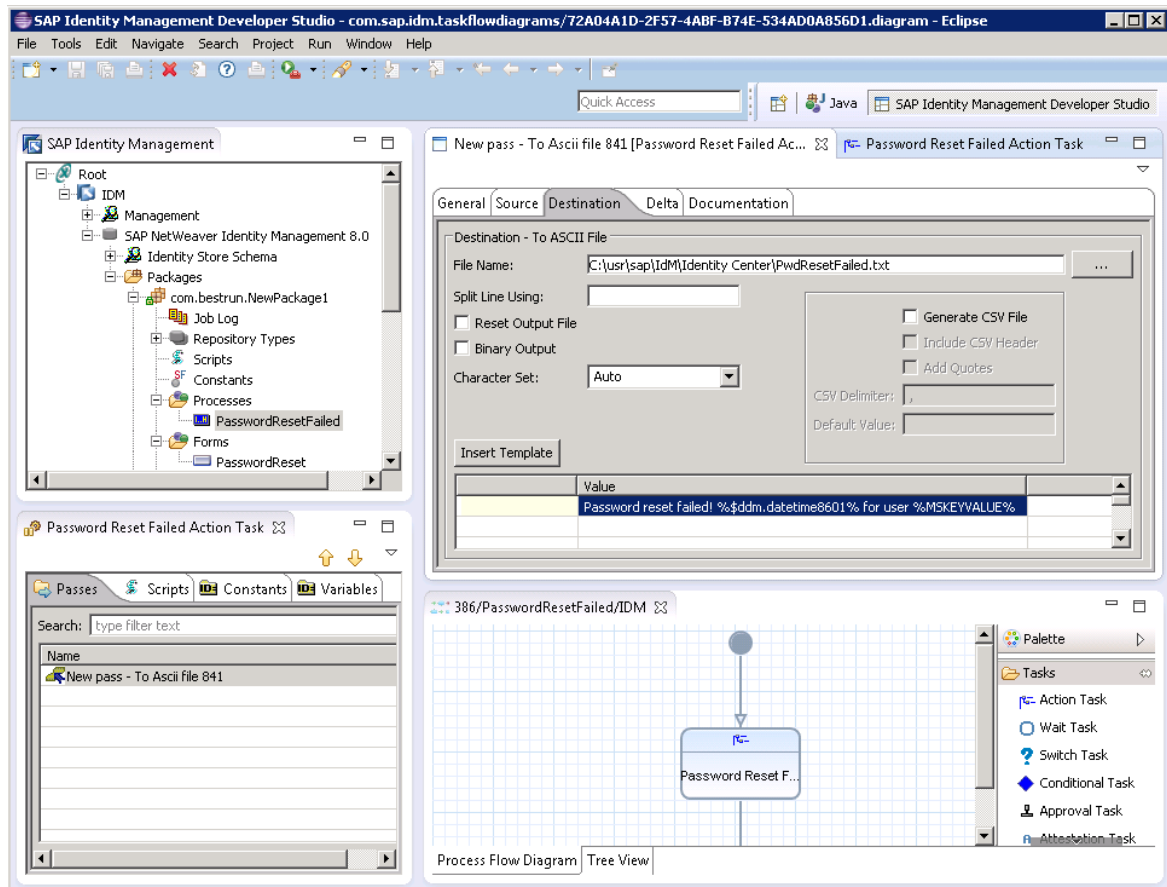
To modify the name of the Action Task, choose **Rename** from the context menu and enter **PasswordResetFailedActionTask**, for example.

8. Select the **PasswordResetFailedActionTask** and choose **Properties** from the context menu (or double-click the task).

9. On the *General* tab, select the *Enabled* checkbox and the dispatcher(s) that are allowed to run this job.
10. In the job view, select the *Passes* tab and choose **New** *To ASCII file* to create a pass.



11. On the *Destination* tab of the pass, modify the following properties:
 - In the *File Name* field, specify the location and the name of the file where the information about the failed password reset process will be available (for example, C:\usr\sap\IdM\Identity Center\PwdResetFailed.txt).
 - In the *Value* field of the definitions table, type any value to be written to the ASCII file (for example, **Password reset failed! %\$ddm.datetime8601% for user %MSKEYVALUE%**. Use the context menu to insert the system parameter `ddm.datetime8601` and the source attribute `MSKEYVALUE`.



12. Choose [Save](#).

3.5.4 Configuring Password Reset Parameters

The password reset form should be configured to call the Password reset failed process. Parameters, like defining a minimum number of validation answers or a number of authentication questions that should be displayed for the user, need to be defined also.

The password reset form should also be referenced from the identity store. This section describes how to configure the parameters on the Password reset form and on the identity store.

Related Information

[Adding Reference to Password Reset Failed Process \[page 570\]](#)

[Setting Password Reset Parameters \[page 571\]](#)

[Adding Reference to Password Reset Form on Identity Store \[page 572\]](#)

3.5.4.1 Adding Reference to Password Reset Failed Process

You need to configure the Password Reset form to call the Password Reset Failed process.

Context

To configure the Password Reset form to call the Password Reset Failed process, proceed as follows:

Procedure

1. In the tree view, select the Password Reset form and choose *Properties* from the context menu.
2. Select the *Parameters* tab and choose ... button to the right of the *Password Reset Failed Task* field.
3. In the *Available processes/forms* dialog box, select the Password Reset Failed process and choose *OK*.

The screenshot shows the 'Parameters' tab of the 'PasswordReset' form configuration. The 'Entry type' is set to '-- None --'. The 'Account Validation' section has a 'Min. no of Validation Answers' field. The 'Identify' section includes 'Description', 'Identification Attribute' (set to 'MSKEYVALUE'), and 'Alternative Identification Attr'. The 'Set Password' section includes 'Description' (set to '#MX_PWDRCV_STEP3'), 'Password Creation Method' (set to 'Automatically Generate'), and a checked 'Save the Password to UME' option. The 'Verify Identity' section includes 'Description', 'No of Questions to Show', 'No of Answers Required', and 'Maximum no of Attempts' (set to 3). The 'Finish' section includes 'Confirmation Message' (set to '#MX_PWDRCV_STEP4') and 'Password Reset Failed Task' (set to '386/PasswordResetFailed').

The reference to the Password Reset Failed process is now added.

4. Choose *Save*.

3.5.4.2 Setting Password Reset Parameters

You need to define password reset parameters, like the minimum number of validation answers or the number of authentication questions that should be displayed for the user.

Context

To set parameters in the Password Reset form, proceed as follows:

Procedure

1. In the tree view, select the Password Reset form and choose *Properties* from the context menu.
2. Select the *Parameters* tab and configure the parameters as described in *Parameters (Password Reset Form)*. See the Related Information section
3. Choose *Save*.

Results

The implementation and configuration of the self-service password reset is completed.

The configuration may look like this:

The screenshot shows the 'PasswordReset' configuration window with the 'Parameters' tab selected. The window has a title bar and a tabbed interface with the following tabs: General, Result Handling, Parameters, Access Control, Presentation, and Documentation. The 'Parameters' tab is active, showing the following configuration fields:

- Entry type:** 6/MX_PERSON
- Account Validation:**
 - Min. no of Validation Answers: 5
- Identify:**
 - Description: TASK_PASSWORDRESET_STEP1
 - Identification Attribute: MSKEYVALUE
 - Alternative Identification Attr: (empty)
- Set Password:**
 - Description: #MX_PWDRCV_STEP3
 - Password Creation Method: Ask the User
 - ☒ Save the Password to UME
- Verify Identity:**
 - Description: #MX_TASK_PASSWORDRESET_STEP2
 - No of Questions to Show: 2
 - No of Answers Required: 2
 - Maximum no of Attempts: 3
- Finish:**
 - Confirmation Message: #MX_PWDRCV_STEP4
 - Password Reset Failed Task: 386/PasswordResetFailed

The next step is to create a self-service user form where the user can define the answers to the authentication questions.

Related Information

[Parameters \(Password Reset Form\) \[page 1097\]](#)

3.5.4.3 Adding Reference to Password Reset Form on Identity Store

You need to configure the identity store to reference the Password Reset form.

Context

To add a reference to Password Reset form on identity store, proceed as follows:

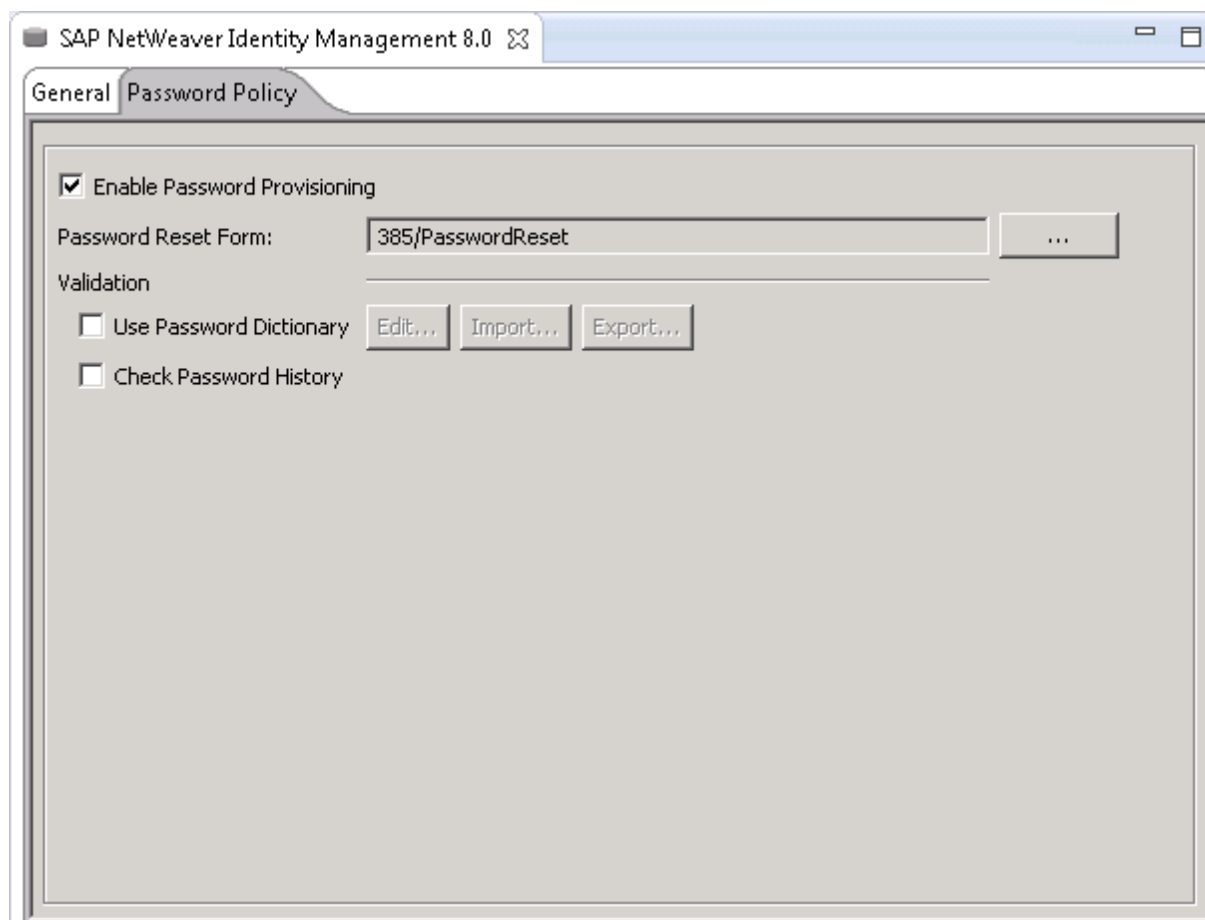
Procedure

1. In the tree view, select your identity store and choose [Properties](#) from the context menu.
2. Select the [Password Policy](#) tab and configure the following:
 - Select the [Enable Password Provisioning](#) checkbox.
 - Select the ... button to the right of the [Password Reset Form](#) field. In the [Available processes/forms](#) dialog box, select the Password Reset form that will be called to perform password reset and choose [OK](#)
 - Enable or disable the use of password dictionary.
 - In the [Check Password History](#) checkbox, define whether the previous passwords should be considered when creating a new one or not. And if yes, how many of the previous passwords should be remembered by the system.
3. Choose [Save](#).

Results

The reference to Password Reset form is now added on the identity store.

The configuration may look like this:



3.5.5 Editing Authentication Information

Now that the self-service password reset is implemented and configured, the user needs to define the answers to the authentication questions. To be able to do that, the user needs a self-service form.

In this document, Edit authentication questions form is created for that purpose, but this can typically be done in a user profile form. That is, if you have defined a self-service form where the users can edit their user profile, add the attributes MX_AUTHQ_001 to MX_AUTHQ_005 to this form.

Related Information

[Creating Edit Authentication Questions Form \[page 574\]](#)

[Defining Answers to Authentication Questions \[page 576\]](#)

3.5.5.1 Creating Edit Authentication Questions Form

Context

To create the self-service form for editing authentication question, proceed as follows:

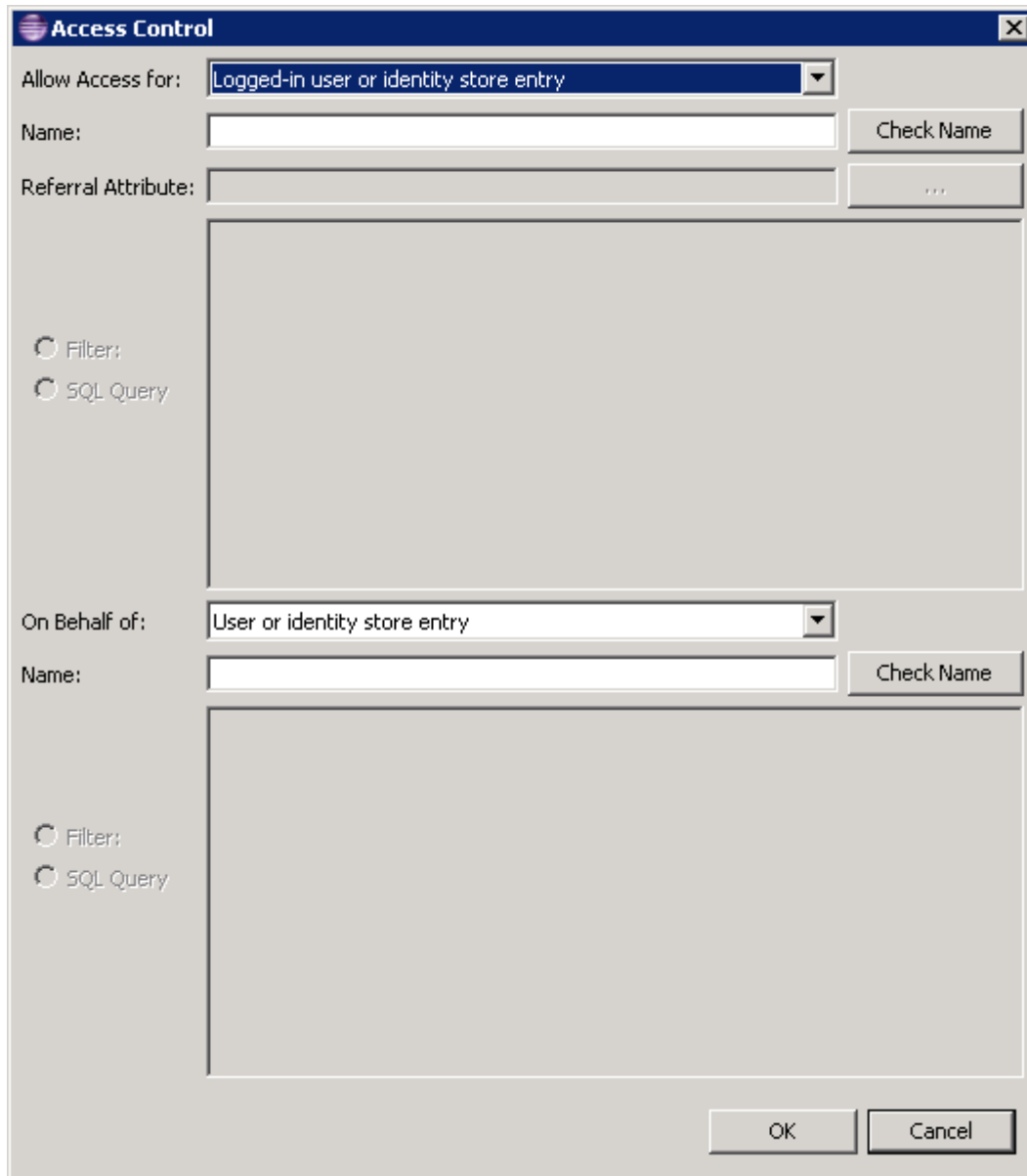
Procedure

1. Select the package where you have created the *PasswordReset* form.
2. Select the *Forms* folder and choose **New > Form**.
3. To modify the name of the form, choose *Rename* from the context menu and enter **Edit Authentication Questions**, for example.
4. Select the *Attributes* tab and in the *Entry Type* field, select MX_PERSON.
5. Select the attributes MX_AUTHQ_001 to MX_AUTHQ_005 and configure the attributes as shown below.

The screenshot shows the SAP Identity Management configuration interface. On the left, a tree view displays the hierarchy: Root > IDM > Management > Job Log > Execution Log > Repositories > Dispatchers > SAP NetWeaver Identity Management 8.0 > Identity Store Schema > Entry Types > Attributes > Packages > com.bestrun.NewPackage1 > Job Log > Repository Types > Scripts > Constants > Processes > Forms > Edit Authentication Questions. The main window is titled '*Edit Authentication Questions' and has tabs for General, Result Handling, Attributes, Access Control, Presentation, and Documentation. The 'Attributes' tab is active. It shows 'Entry Type' as '6/MX_PERSON'. Below, a table lists 'Form Attributes' with columns for Attribute, List, Mandatory, and Read Only. The attributes are: DISPLAYNAME, MSKEYVALUE, MX_AUTHQ_001, MX_AUTHQ_002, MX_AUTHQ_003, MX_AUTHQ_004, MX_AUTHQ_005, DESCRIPTION, MX_ACADEMIC_TITLE_1, MX_ACADEMIC_TITLE_2, MX_ACCESSIBILITYLEVEL, MX_ACCOUNTING_NUMBER, and MX_ADDRESS_CITY. The 'List' column is checked for all attributes. The 'Mandatory' column is checked for DISPLAYNAME, MSKEYVALUE, and MX_AUTHQ_001. The 'Read Only' column is checked for DISPLAYNAME, MSKEYVALUE, and MX_AUTHQ_001. There are 'Up' and 'Down' buttons on the right side of the table.

Attribute	List	Mandatory	Read Only
DISPLAYNAME	☒	☒	☒
MSKEYVALUE	☒	☒	☒
MX_AUTHQ_001	☒	☒	☒
MX_AUTHQ_002	☒	☐	☐
MX_AUTHQ_003	☒	☐	☐
MX_AUTHQ_004	☒	☐	☐
MX_AUTHQ_005	☒	☐	☐
DESCRIPTION	☐	☐	☐
MX_ACADEMIC_TITLE_1	☐	☐	☐
MX_ACADEMIC_TITLE_2	☐	☐	☐
MX_ACCESSIBILITYLEVEL	☐	☐	☐
MX_ACCOUNTING_NUMBER	☐	☐	☐
MX_ADDRESS_CITY	☐	☐	☐

6. Select the *Access Control* tab and choose *Add*.
7. In the *Access Control* dialog box, provide the following information:
 - In the *Allow Access for* field, select *Logged-in user or identity store entry*.
 - In the *Name* field, you have two options:
 - To make the form accessible to everyone, leave this field empty.
 - To restrict the access to this form, enter a name. For example, enter Administrator name to give access to this form only to the Administrator user.
 - In the *On Behalf of* field, either select *User or identity store entity* or *Relation - Self*. Both ways are legitimate.



The image shows a dialog box titled "Access Control" with a close button (X) in the top right corner. The dialog is divided into two main sections, each with a "Name:" label and a "Check Name" button. The top section has a dropdown menu labeled "Allow Access for:" with the value "Logged-in user or identity store entry". Below this is a "Referral Attribute:" label and a button with three dots "...". The bottom section has a dropdown menu labeled "On Behalf of:" with the value "User or identity store entry". Both sections have a "Filter:" radio button and an "SQL Query" radio button, followed by a large empty text area. At the bottom right, there are "OK" and "Cancel" buttons.

Access Control

Allow Access for: Logged-in user or identity store entry

Name: Check Name

Referral Attribute: ...

☐ Filter:
☐ SQL Query

On Behalf of: User or identity store entry

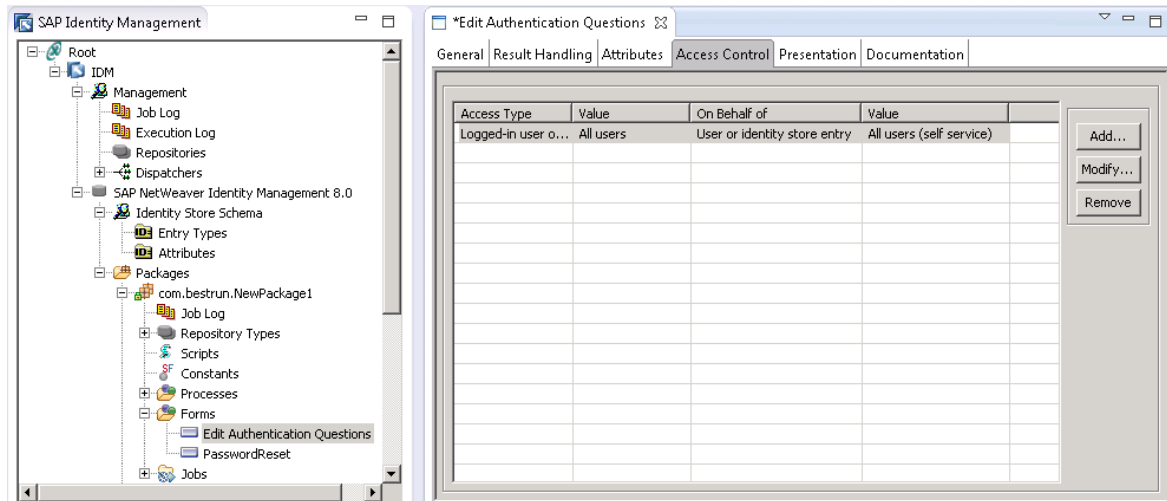
Name: Check Name

☐ Filter:
☐ SQL Query

OK Cancel

8. Choose [OK](#).

The access definition is added to the form.



9. Choose [Save](#).

3.5.5.2 Defining Answers to Authentication Questions

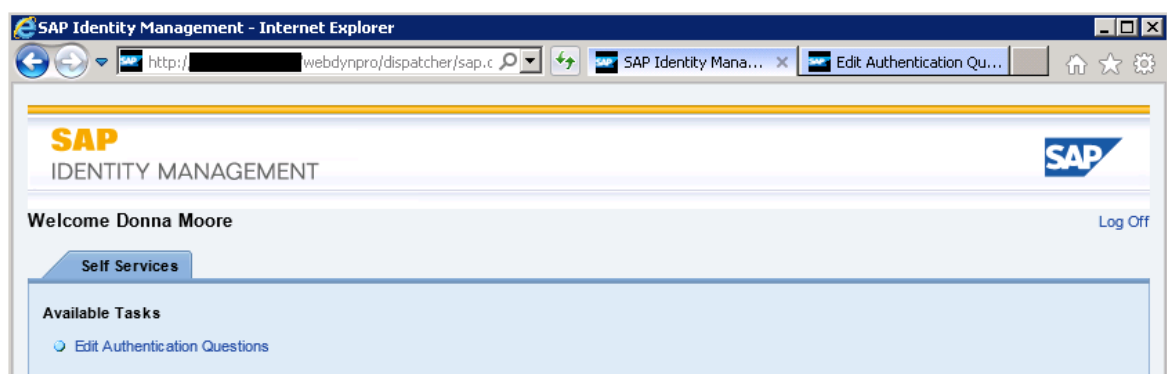
To define the answers to the authentication questions or to edit the answers, the user needs to access the self-service form Edit Authentication Questions.

Context

Proceed with the following:

Procedure

1. Access SAP Identity Management User Interface at <http://<host>:<port>/idm> and provide the credentials in the log-in window.
2. Choose [Log On](#).



You are now logged on to the User Interface. The image above shows the logged-in user with access to the [Self Services](#) tab and the self-service form [Edit Authentication Questions](#).

3. Select the [Edit Authentication Questions](#) form and provide the answers to the displayed authentication questions.

Note

The answers of the authentication questions are only known by the user, and not available from observation, public records, or social media.

4. Choose [Save](#).

Edit Authentication Questions

Unique ID: DonnaMoore
Display Name: Donna Moore

☒ The task has been executed

[Save](#) [Modify](#) [Refresh](#)

Display Name: Donna Moore
Unique ID: DonnaMoore
What is your favorite color?:
What make of car do you drive?:
What is your pet's name?:
What is your mother's maiden name?:
What street did you grow up on?:

The answers only known by the user (in this case Donna Moore), are now defined for the authentication questions.

Note

If you try to save the information after entering the answers to fewer questions than requested, the form will fail and an error message will appear, as displayed below.

Edit Authentication Questions

Unique ID: DonnaMoore
Display Name: Donna Moore

You must answer at least 5 validation question(s).

[Save](#) [Modify](#) [Refresh](#)

Display Name: Donna Moore
Unique ID: DonnaMoore
What is your favorite color?:
What make of car do you drive?:
What is your pet's name?:
What is your mother's maiden name?:
What street did you grow up on?:

3.5.6 Testing Self-Service Password Reset

The password reset process is now defined and the self-service password reset functionality is ready for use. In this section, we will test the self-service password reset functionality, that is, the Password Reset form and the Password Reset Failed action task.

To access the Password Reset form, make sure to be logged out of the identity store. Make also sure that all browser windows are closed.

📘 Note

In what language the self-service password reset is available, is determined in the following way:

- The language is determined by the language preferences specified by the browser settings ("accept-language" HTTP header), if existing.
- Otherwise, the language is determined by the language preferences specified in the application properties, if existing.
- Otherwise, the language is determined by the language preferences specified in the Web Dynpro system properties, if existing.
- Otherwise, the language is determined by the language preferences of the VM.

Related Information

[Providing New Password \[page 578\]](#)

[Testing Password Reset Failed Action Task \[page 580\]](#)

3.5.6.1 Providing New Password

Context

To access the self-service password reset functionality and reset your password, proceed as follows:

Procedure

1. Open your browser and enter `http://<host>:<port>/idm/pwdreset` to access the password reset self-service form.

Note

You may wish to incorporate the password reset link to the logon screens (welcome pages) of application servers accessed by users (assuming these are serviced by Identity Management).

For example, if your users access an SAP NetWeaver Application Server Java, then you can customize the logon screen for that host to access the password reset URL of SAP NetWeaver Identity Management. For more information, see the SAP Help Portal documentation relevant for your AS Java release. For SAP NetWeaver 7.3, SAP NetWeaver 7.3 EHP 1 and SAP NetWeaver 7.4, see [Developing a Custom Logon Screen](#).

2. Enter the unique ID (here MSKEYVALUE as previously defined) and choose [Next](#).

SAP Identity Management - Internet Explorer

http://[redacted]webdynpro/dispatcher/sap.com/!

SAP Identity Management

PasswordReset

Help

Previous Next

1 Identify 2 Verify Identity 3 Set Password

Task_PASSWORDRESET

Unique ID: DonnaMoore

3. Enter the answers to the authentication questions and choose [Next](#).

SAP Identity Management - Internet Explorer

http://[redacted]webdynpro/dispatcher/sap.com/!

SAP Identity Management

PasswordReset

Help

Previous Next

1 Identify 2 Verify Identity 3 Set Password

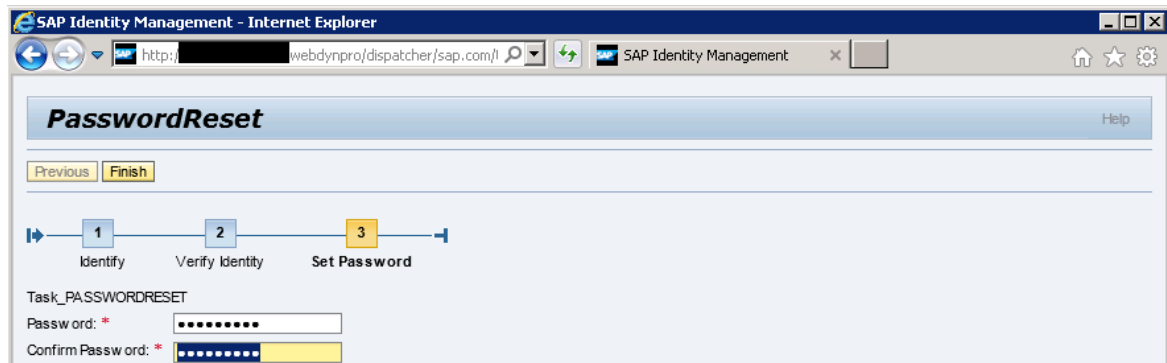
Task_PASSWORDRESET

What make of car do you drive?:

What street did you grow up on?:

If the authentication succeeds, the user can set the new password. In this document, the method [Ask the user](#) is used for password creation, giving the user the possibility of defining its own password.

4. Enter the new password and confirm it.



5. Choose *Finish* to set the password.

The password is now set.

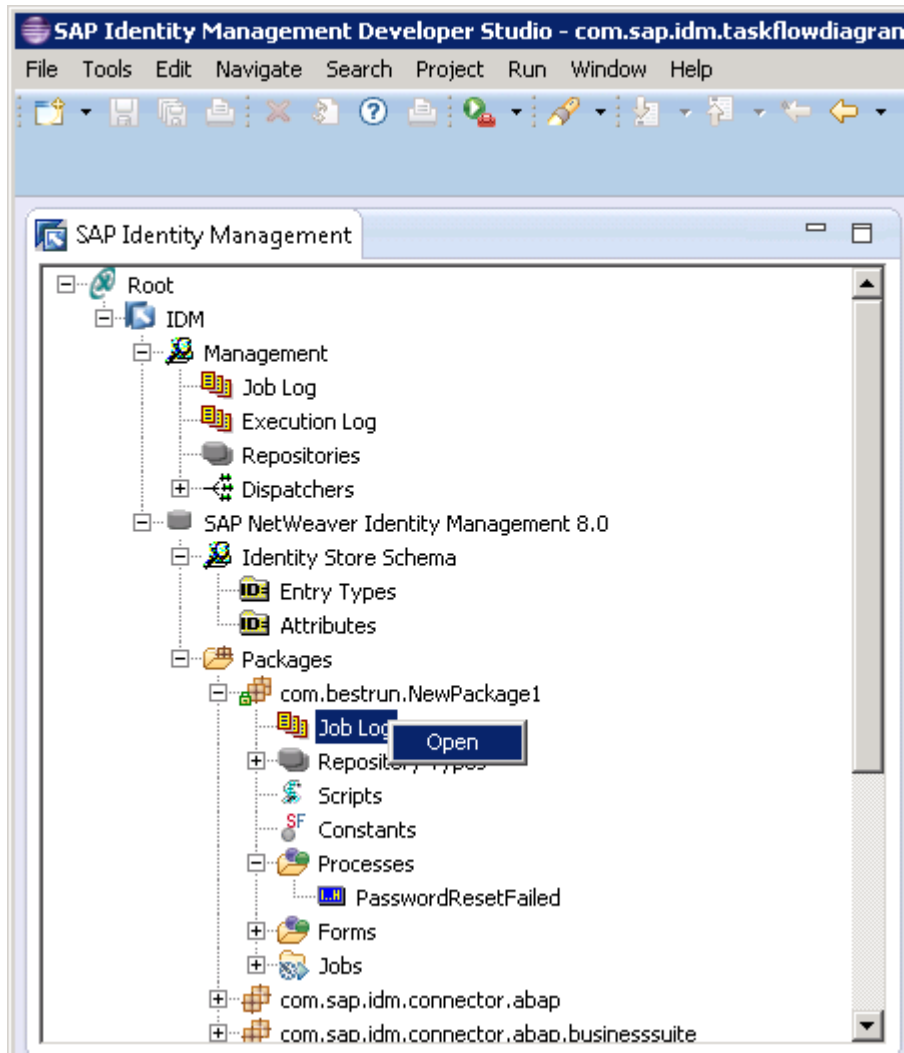
3.5.6.2 Testing Password Reset Failed Action Task

Context

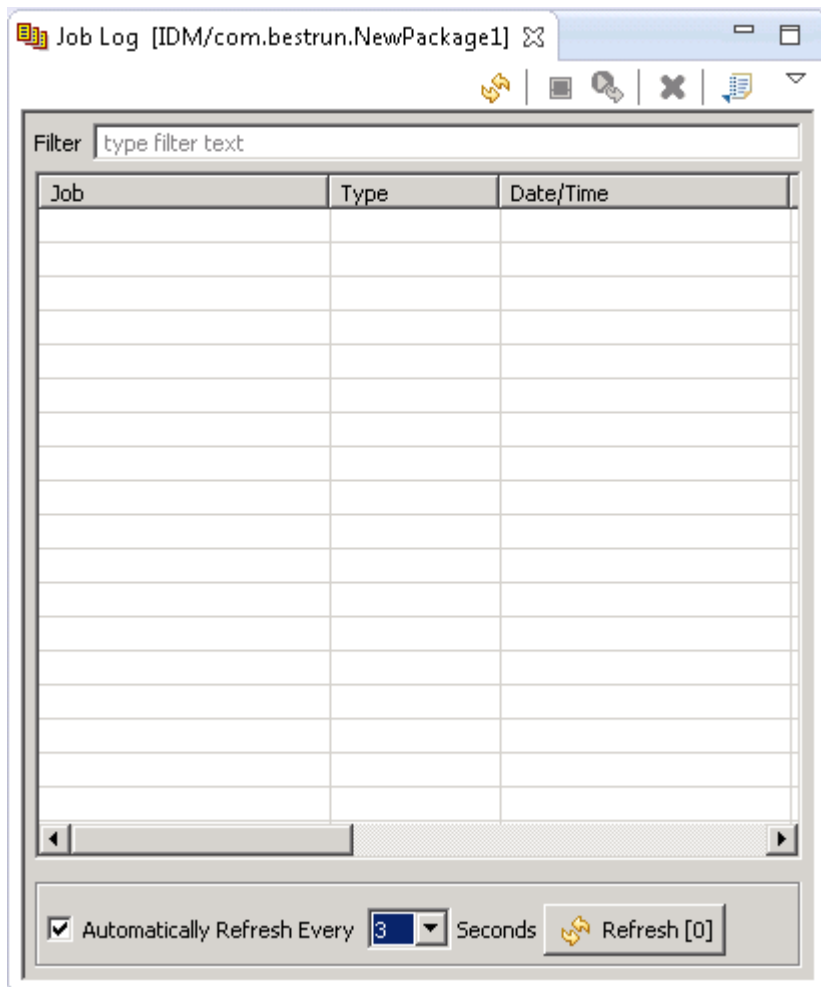
To test and verify that the Password Reset Failed action task works, proceed as follows:

Procedure

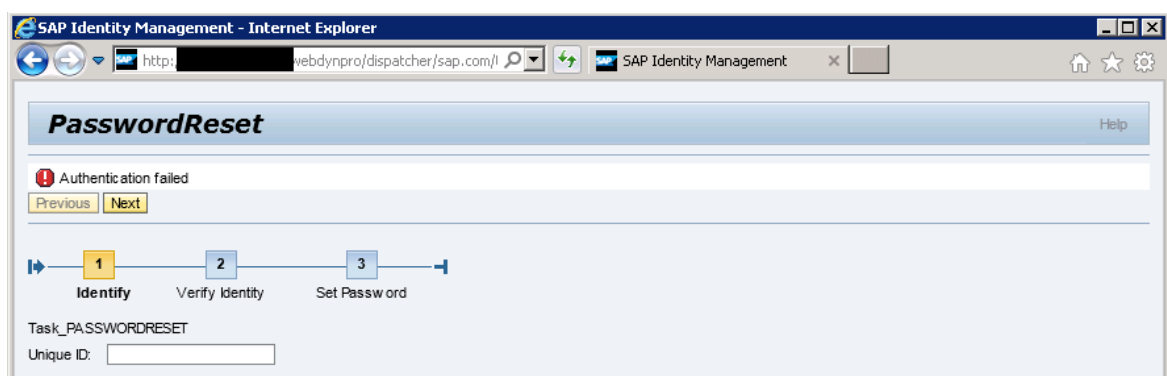
1. Navigate to and select the *Job Log* of the package where you have created the Password Reset Failed process that contains the Password Reset Failed action task.



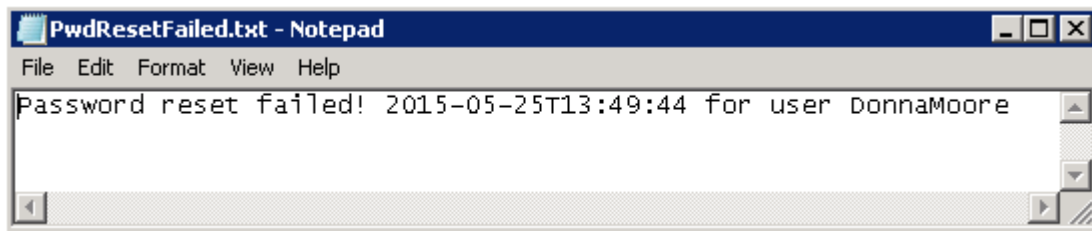
2. In the *Job Log* view, select the *Automatically Refresh Every <n>Seconds* checkbox and set it to 3 seconds.



3. Open your browser and enter `http://<host>:<port>/idm/pwdreset` to access the password reset self-service form.
4. Enter the unique ID and then enter wrong credentials (for example, the wrong answers to the authentication questions) to cause authentication error.



5. Inspect the [Job Log](#) to verify that the job has run without errors.
6. In addition, check the defined location of the file to verify that the file was created with the expected input. The file may look like this:



Note

Every failed attempt of password reset is logged. For security reasons the user is not told why a password reset attempt failed, if too many attempts are made to reset the password or if the provided unique identifier does not exist, as this would provide a potential attacker with additional information. Instead, the password reset will proceed but random authentication questions may be displayed to the user (including the ones that user has not defined the answers for) and the password reset will fail regardless of the input information being correct or not.

Note

Aborting the self-service password reset (for example, by closing the browser window) while in Step 2 (Verify Identity) or Step 3 (Set Password) will count as a failed password reset attempt, but this type of failed password reset attempt will not be recognized by the Password Reset Failed action task and thus not be logged to the specified file.

3.5.7 Changing Authentication Questions

You can change the authentication questions by changing the display name of attributes MX_AUTHQ_001 to MX_AUTHQ_005.

Context

Changing the authentication questions should be done during the implementation of the self-service password reset. That is, before the form allowing the users to enter answers to the authentication questions is available for the users. Changing the questions after the users have provided their answers will cause the answers not to fit well any more.

By default, the authentication questions are:

- What is your favorite color? (MX_AUTHQ_001)
- What make of car do you drive? (MX_AUTHQ_002)
- What is your pet's name? (MX_AUTHQ_003)
- What is your mother's maiden name? (MX_AUTHQ_004)
- What street did you grow up on? (MX_AUTHQ_005)

Note

The display names of the authentication questions are determined by the language dependable text keys #MX_MX_AUTHQ_001_DN to #MX_MX_AUTHQ_005_DN available for a set of languages. They are

presented in language, selected for your Identity Center. Changing the display name of the authentication question will replace this language dependable text key with your own text key with a value of your choice in a language of your choice.

To change the authentication questions, proceed as follows:

Procedure

1. In the tree view, navigate to and select *Attributes* for your identity store.
All attributes in your identity store are listed.
2. Navigate to the attribute you want to change, for example MX_AUTHQ_001 attribute, and select the *Presentation* tab:

The screenshot shows a configuration window titled 'MX_AUTHQ_001' with several tabs: General, Storage, Presentation, Validation, Event Processes, Attribute Values, and Entry Types. The 'Presentation' tab is selected. The configuration fields are as follows:

- Display Name:** A text field containing '#MX_MX_AUTHQ_001_DN' and a button with three dots.
- Tooltip Text:** An empty text field and a button with three dots.
- Presentation:** A dropdown menu currently showing 'SingleLine'.
- Options:** A group of checkboxes: 'Confirm Input' (unchecked), 'Hide Input' (checked), 'Show Search Field' (unchecked), and 'List Entries on Load' (checked).
- Number of Lines:** A text field containing the value '5'.
- Task Defaults:** A group of checkboxes: 'Read Only' (unchecked) and 'Mandatory' (unchecked).

3. In the *Display Name* field, alter the displayed text key to (make your own), for example **#MX_MX_AUTHQ_001_DN_MY**.

The screenshot shows a configuration window titled '*MX_AUTHQ_001' with several tabs: General, Storage, Presentation, Validation, Event Processes, Attribute Values, and Entry Types. The 'Presentation' tab is active. It contains the following fields and options:

- Display Name:** A text field containing '#MX_MX_AUTHQ_001_DN_MY' and a button with three dots '...' to its right.
- Tooltip Text:** An empty text field and a button with three dots '...' to its right.
- Presentation:** A dropdown menu currently showing 'SingleLine'.
- Options:**
 - ☐ Confirm Input
 - ☒ Hide Input
 - ☐ Show Search Field
 - ☒ List Entries on Load
- Number of Lines:** A text field containing the value '5'.
- Task Defaults:**
 - ☐ Read Only
 - ☐ Mandatory

4. Choose the ... button to the right of the *Display Name* field to define the values for the new text key.
5. In the *Localize Text* dialog box, change the authentication question for the languages you prefer (in this example the change is defined for English).

Localize Text [X]

Text Key: #MX_MX_AUTHQ_001_DN_MY

Language	Text
AR	
BG	
CA	
CS	
DA	
DE	
EL	
EN	What is your favorite drink?
ES	
ET	
FI	
FR	
HE	
HR	
HU	
IT	
JA	
KO	
LT	
LV	
MX	
NL	
NO	
PL	
PT	
RO	
RU	
SH	
SK	
SL	
SV	
TH	
TR	
UK	
VI	
ZH_CN	
ZH_HK	
ZH_TW	

OK Cancel

6. Choose **OK**.

You have now changed the question from **What is your favorite color?** to **What is your favorite drink?**. You can observe the change in the self-service form where the user defines the answers to the authentication questions.

The change is also visible in the password reset form.

3.5.8 Resetting Number of Failed Password Reset Attempts

If a user has exceeded the maximum number of failed password reset attempts, the number must be reset (cleared). You need to create a form for setting password for the user and a process for resetting the number of attempts. Then you need to specify the process to be executed if the form completes successfully.

Context

Proceed as follows:

Procedure

1. In the tree view, select the *Forms* folder and choose **New > Form**.
To modify the name of the form, choose *Rename* from the context menu and enter **Set Password and Reset Attempts Counter**, for example.
2. Select the *Attributes* tab and in the *Entry Type* field, select MX_PERSON.
3. Configure the attributes for this form. Use *Up* or *Down* to place the attributes in the exact same order as shown below.

Set Password and Reset Attempts Counter

General | Result Handling | **Attributes** | Access Control | Presentation | Documentation

Entry Type: 6/MX_PERSON ... ☐ Report Fo
☐ This Form

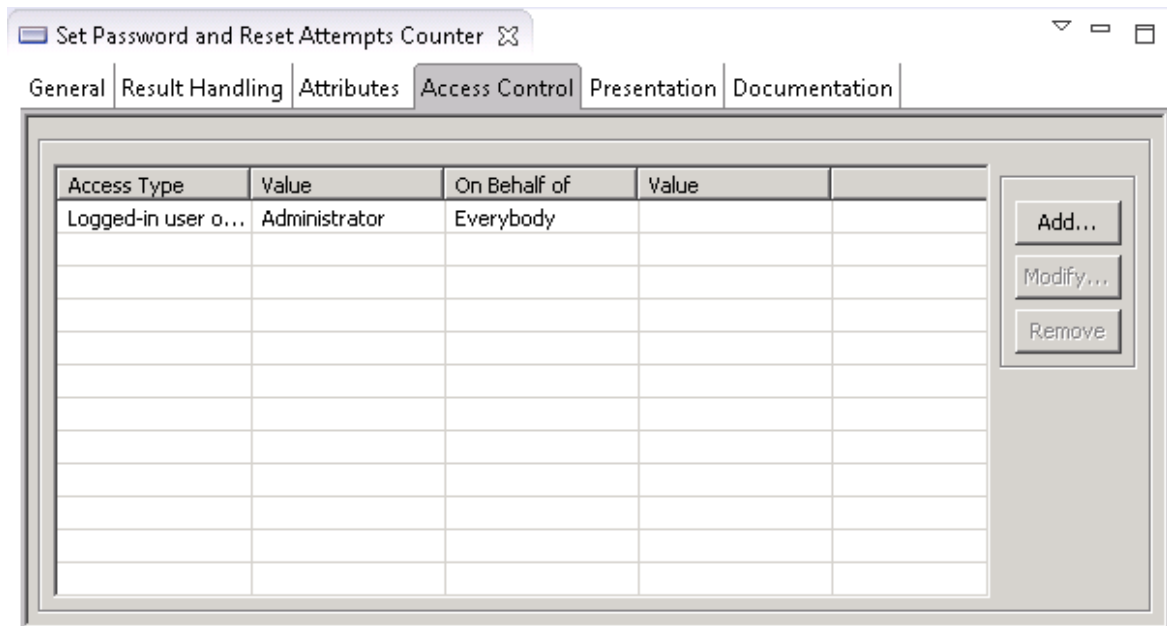
Form Attributes:

Attribute	List	Mandatory	Read
MSKEYVALUE	☒	☒	☒
DISPLAYNAME	☒	☒	☒
MX_PASSWORD	☒	☐	☐

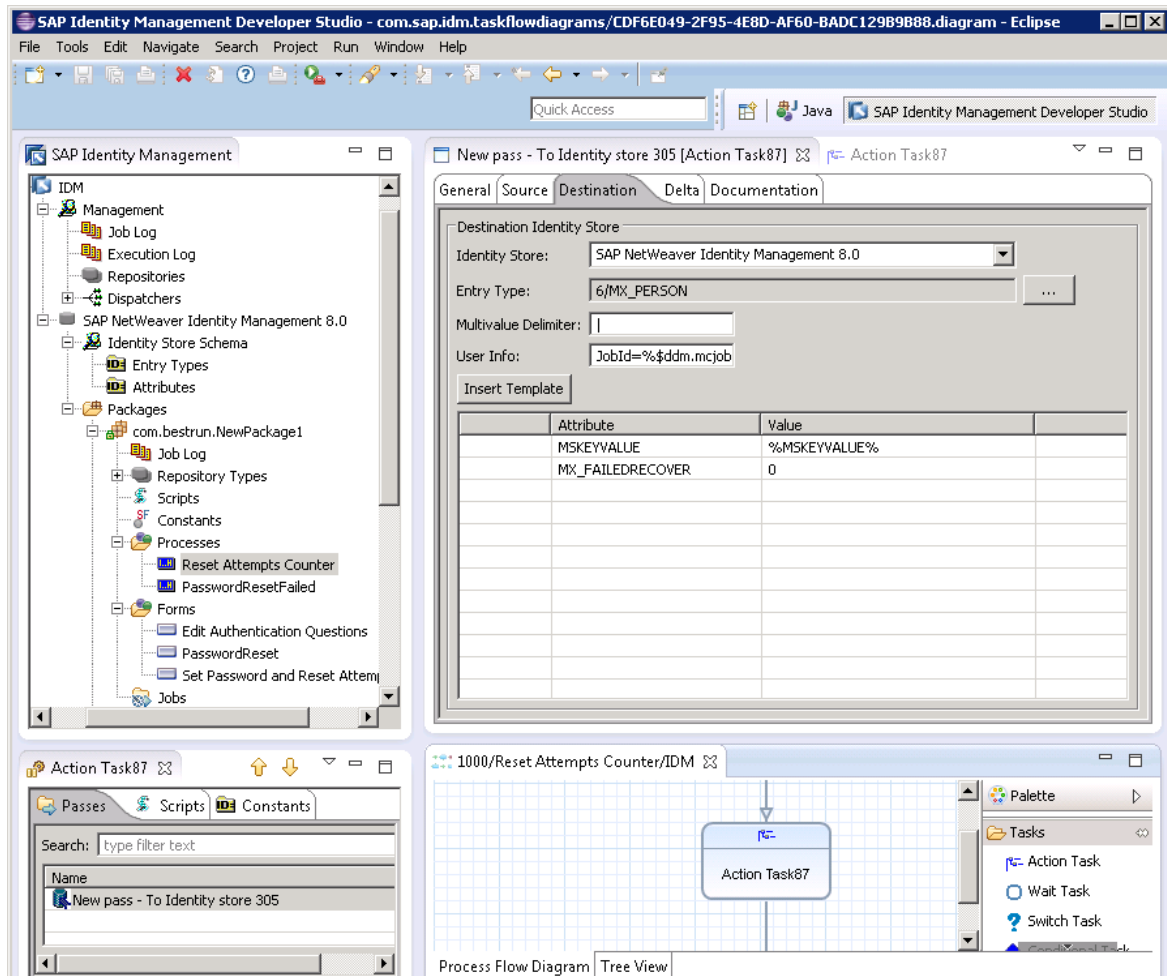
Up
Down

4. Select the [Access Control](#) tab and choose [Add](#).
5. In the [Access Control](#) dialog box, configure the following:
 - In the [Allow Access for](#) field, select [Logged-in user or identity store entry](#).
 - In the [Name](#) field, enter the name of the identity store user with access to the [Manage](#) tab in the User Interface (here **Administrator**). You might use [Check Name](#) to ensure that the name you entered is correct and exists. This allows the administrator user access to the form.
 - In the [On Behalf of](#) field, select [Everybody](#).
 - Choose [OK](#).

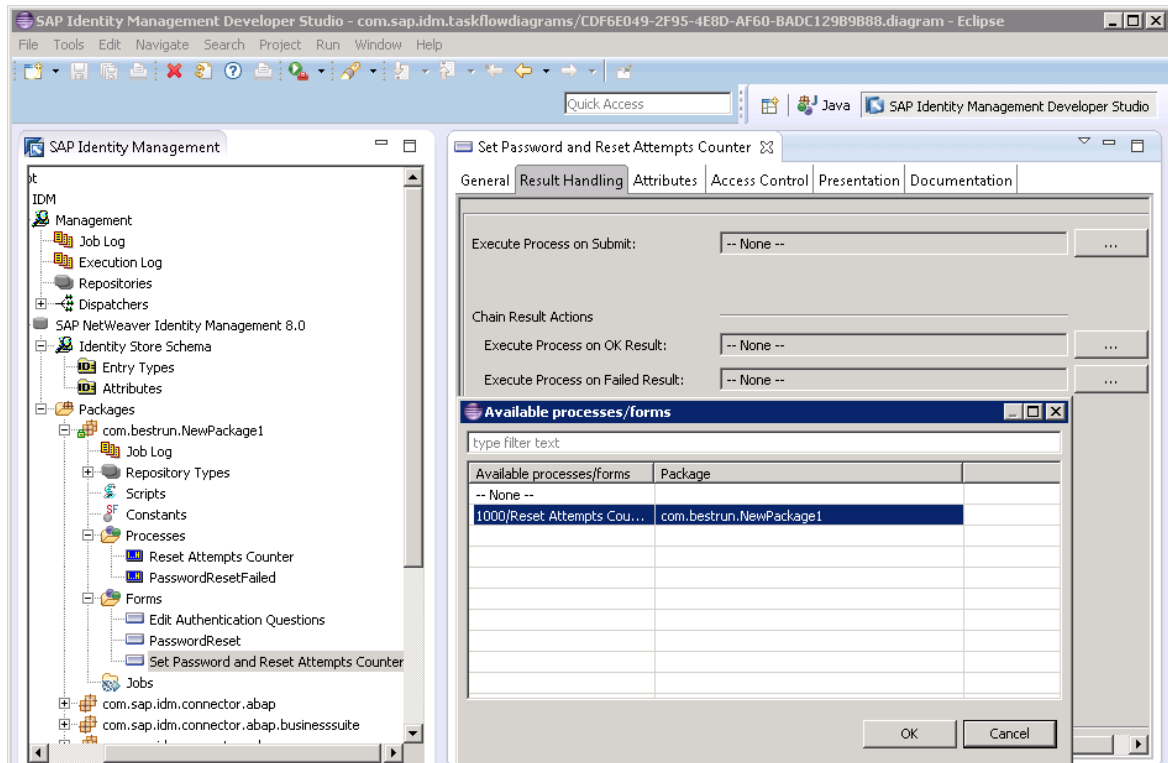
The access definition is added to the form.



6. Choose *Save*.
7. In the tree view, select the *Processes* folder and choose ► *New* ► *Process* .
To modify the name of the process, choose *Rename* from the context menu and enter **Reset Attempts Counter**, for example.
8. On the *General* tab of the Reset Attempts Counter process, define the type of the process to be *Result Handling OK Process*.
9. In the process flow diagram, select *Action Task* from the palette to the right of the canvas and add the task to the Reset Attempts Counter process.
10. Open the Action Task properties and then on the *General* tab, select the *Enabled* checkbox and the dispatcher(s) that are allowed to run this job.
11. In the job view, select the *Passes* tab and choose ► *New* ► *To Identity Store* pass.
12. On the *Destination* tab of the pass, modify the following properties:
 - In the *Identity Store* field, make sure the correct identity store is selected.
 - In the *Entry Type* field, select MX_PERSON.
 - In the definitions table, add the attributes, as shown below. You can use the context menu to insert the attributes.



13. Choose **Save**.
14. Specify the **Reset Attempts Counter** process to be executed if the **Set Password and Reset Attempts Counter** form completes successfully. Open the form properties and select the **Result Handling** tab.
15. Select the ... button to the right of **Execute Process on OK Result** field. Select the **Reset Attempts Counter** process and choose **OK**.



16. Choose [Save](#).

Results

The form is now created. When you log on to the User Interface, the form will look like this:

The password can be set for the user and the number of failed attempts will be automatically cleared for the given user when the form is saved (that is, the Reset attempts counter process will be executed).

3.6 Configuring E-Mail Notifications

You can configure e-mail notification to send notification messages from the SAP provisioning framework, approval and attestation tasks.

Context

To use e-mail notifications, you need the notification package `com.sap.idm.util.notification` provided with the Notification framework. This package contains the notification process and the notification templates that are used to send notifications from the SAP provisioning framework, approval and attestation tasks.

Configuring e-mail notifications depends on the type of notifications you want to send, whether as a result of SAP provisioning framework processes or from approval or attestation tasks.

Notifications from SAP Provisioning Framework Processes

Procedure

1. In SAP Identity Management Developer Studio, import the notification package.
2. In SAP Identity Management Administration User Interface, verify that MX_SAP_PF templates are available. Navigate to ► [Message Templates](#) ► [Show Category](#) ► [MX_SAP_PF](#) ► [Go](#) .
3. In SAP Identity Management Administration User Interface, open the notification package. Navigate to ► [System Configuration](#) ► [Packages](#) ► [com.sap.idm.util.notification](#) .
4. Configure the package constants. See [Notification Package \[page 477\]](#)

To trigger a notification as a result of an SAP Provisioning Framework process, the value of the NOTIFYEVENT constants should point to the template you want to use.

For example: If you want to send a notification for a completed privilege assignment, in the [Value](#) field of the NOTIFYEVENT_ASSIGNMENT_COMPLETED constant, enter **PF Privilege assignment notification** .

Results

A privilege is assigned to a user with a valid e-mail address. When the assignment is provisioned, the user receives e-mail notification.

Notifications from Approval or Attestation Tasks

Procedure

1. In SAP Identity Management Developer Studio, import the notification package.
 2. In SAP Identity Management Administration User Interface, verify that MX_APPROVALS and MX_ATTESTATIONS templates are available. Navigate to ► [Message Templates](#) ► [Show Category](#) ►.
 - For approval templates, select [MX_APPROVALS](#) and choose [Go](#).
 - For attestation templates, select [MX_ATTESTATIONS](#) and choose [Go](#).
 3. In SAP Identity Management Administration User Interface, open the notification package. Navigate to ► [System Configuration](#) ► [Packages](#) ► [com.sap.idm.util.notification](#) ►.
 4. Configure the package constants. See [Notification Package \[page 477\]](#)
 5. In SAP Identity Management Developer Studio, create a public process and add an approval or attestation task.
 6. Configure the approval or attestation task and add the notification process. See [Configuring the Approval \[page 257\]](#) and [Configuring the Attestation \[page 267\]](#)
 - To trigger a notification from an approval task, on the [Approval](#) tab, select the template you want to use in the [Notifications](#) section fields.
 - To trigger a notification from an attestation task, on the [Attestation](#) tab, select the template you want to use in the [Notifications](#) section fields.
- For example: If you want to send the initial message to the approver, in the [Initial Message](#) field, select [Approval initial notification](#) template.
- If you get approvers or attestors from a role or a privilege, in SAP Identity Management User Interface, you need to add the public process as a [Validate Add Task](#) on the [Member Events](#) tab of the role or the privilege.
7. Save your configuration.

Results

The approver receives notification (for example, an assignment is requested for a user). The approver approves the assignment request on the [To Do](#) tab in SAP Identity Management User Interface. The assignment is provisioned. If notifications from SAP Provisioning Framework processes is also configured, the user receives notification that the assignment request was approved.

If any problems related to e-mail notifications occur, see *E-Mail Notifications are not Triggered* in the *Troubleshooting* section of the *SAP Identity Management Solution Operation Guide*.

Related Information

[E-Mail Notifications are not Triggered](#)

4 Connecting SAP Systems

This section describes how to connect SAP systems to SAP Identity Management and set up your landscape accordingly.

Related Information

[Setting Up an SAP HCM System \[page 602\]](#)

[Setting Up an AS ABAP System \[page 623\]](#)

[Setting Up an AS Java System \[page 628\]](#)

[Setting Up an SAP HANA System \[page 632\]](#)

[Setting Up a SuccessFactors System \[page 640\]](#)

4.1 Determine the System Landscape

Once you have set up the initial configuration, you must set up the connection for your particular system landscape. You can use the information provided in the *Use Cases* section to determine which use case best fits to your system landscape. This can help when determining which system you use as the leading identity system. See the table below.

Use Case Overview

Use Case	Leading Identity System	Source System for Data	Provisioned Data
SuccessFactors Employee Central (SFSF EC) Integration	SFSF EC	SFSF EC: Employee data (Identities)	LDAP server: Users and user/group assignments AS Java: UME users, UME groups and Role assignments AS ABAP: Users, user/role assignments, and user/profile assignments
SAP HCM	SAP HCM	SAP HCM: Employee data (Identities)	SAP HCM: Employee data (Identities) LDAP server: Users and user/group assignments

Use Case	Leading Identity System	Source System for Data	Provisioned Data
SAP Enterprise Portal	Corporate LDAP directory	LDAP server: Users and groups	AS Java (read from LDAP): UME users and UME groups
		AS Java: Portal roles, UME roles	AS Java (provisioned from IC): Role assignments
		AS ABAP: ABAP roles, ABAP profiles, company addresses	AS ABAP: Users, user/role assignments, and user/profile assignments
Identity Lifecycle Management	SAP HCM	SAP HCM: Employee data (Identities)	LDAP server: Users and user/group assignments
		AS Java: Portal roles and UME roles	AS Java (read from LDAP): UME users and UME groups
		AS ABAP: ABAP roles, ABAP profiles, company addresses	AS Java (provisioned from IC): Role assignments AS ABAP: Users, user/role assignments, and user/profile assignments
Enhanced SAP Business Suite Integration	SAP HCM	SAP HCM: Employee data (Identities)	SAP HCM: Employee data (Identities) LDAP server: Users and user/group assignments SAP Business Suite systems: Users, user/role assignments, and user/profile assignments plus application-specific identity data according to SAP Business Suite scenario.
Central User Administration (CUA)	Central User Administration (CUA)	Identity Center	SAP Identity Management Identity data

Note

Keep in mind that when setting up the portal environment use case, the corporate LDAP directory is used as both the leading system for SAP Identity Management and as the user data store for the portal system. Therefore, set up the Identity Center to read the user data from the LDAP directory server and provision users to all of the target systems except for the AS Java that uses the LDAP directory server as its data source. This AS Java/portal system reads the user data directly from the LDAP directory server.

User/role assignments are provisioned to all systems, including the AS Java where the portal runs.

Based on this information:

1. Identify your leading identity system.

2. Identify your target systems.
3. Continue with setting up the landscape as described in the sections that follow.

Related Information

[Use Cases \[page 19\]](#)

[Setting up the Landscape \[page 596\]](#)

4.2 Setting up the Landscape

Context

Once you have identified the systems you want to connect to SAP Identity Management, set up the landscape accordingly. See *Restrictions and Considerations* for considerations that apply to each use case and each connector type.

Proceed as follows:

Procedure

1. Create repositories for each of the systems.
2. For AS ABAP systems, read the value help content from the system(s) that provide the value help for attributes.
3. If you are using an SAP HCM system as the leading system for employee data, then set it up accordingly. If you are not using SAP HCM in your landscape, you can skip this step.
4. Create and configure the jobs needed for each connector.
5. Prepare the initial loads.
6. Run the initial loads for each connected system.
7. Clean up the collected data.

Related Information

[Restrictions \[page 1330\]](#)

[Creating Repositories \[page 493\]](#)

[Reading Value Help Content \[page 598\]](#)

[Setting Up an SAP HCM System \[page 602\]](#)

[Preparing the Initial Loads \[page 663\]](#)

[Running the Initial Loads \[page 667\]](#)

4.2.1 Attributes Using Value Help Content

There are certain attributes in the provisioning framework's identity store schema that make use of the value help provided by the AS ABAP system(s).

These attributes are classified into three categories, according to the storage location:

- Fixed values: The value help content is maintained in fixed domain values or data elements. Examples for this category include value help for date or number formats.
- System tables: The value help content is maintained in ABAP system tables (not modifiable). Examples for this category include value help for languages or time zones.
- Customer table entries: The value help content is available in tables that are modifiable by customers, for example, application tables, control tables, or Customizing tables. Examples for this category include value help for printer settings or salutations.

Note

For a complete list of attributes that support value help, see *Appendix C: Attributes that Support Value Help*.

The value help content that is either fixed or available in system tables is delivered with the provisioning framework for SAP systems and stored in the `mx_i_AttrValueHelp` database table. The value help content that you can define yourself must be read from the corresponding AS ABAP system into this table. To do this, use the job *ABAP Read Help Values*.

Caution

If there are discrepancies between the values maintained in the `mx_i_AttrValueHelp` table and the data read from the system during an initial load, you will receive errors.

Therefore, if you are connecting multiple AS ABAP systems to the Identity Center that all use value help, then choose one of these systems as the leading system for the value help for each attribute. Make sure the value help content is maintained in this system so that it can be used by all of the connected systems.

Caution

If you do not read the value help (or maintain their values in the identity store schema manually), then these values are missing in the table and cannot be found for the corresponding attributes. In this case, you will receive errors during the initial loads.

Related Information

[Attributes that Support Value Help \[page 1311\]](#)

4.2.2 Reading Value Help Content

Prerequisites

You know which system is the leading system to use for the value help for each attribute.

Context

Follow the steps below:

Procedure

1. Set the properties for each attribute in the identity store schema that uses value help.
For those attributes where the data is provided with the provisioning framework for SAP systems (as fixed values or in system tables), the properties should be set correctly. You can use the procedure below to check these properties.
2. Run the job that reads the value help from the leading system.

Related Information

[Changing the Properties for Each Attribute \[page 599\]](#)

[Running the Read Help Values Job \[page 601\]](#)

4.2.2.1 Changing the Properties for Each Attribute

Context

To change the properties for an attribute, do the following:

Procedure

1. In the Identity Management Developer Studio, navigate to ► *Identity Store Schema* ► *Attributes* ►.
2. Select the attribute to modify with a double-click. (See the table in *Appendix C: Attributes that Support Value Help*.
The properties for the attribute appear.
3. Choose the *Presentation* tab page.
4. Set the *Presentation* option to *ObjectValueHelp*.

See the example below for the MX_SALUTATION attribute.

The screenshot shows the 'Presentation' tab of the attribute configuration dialog. The 'Display Name' is '#MX_MX_SALUTATION_DN'. The 'Presentation' dropdown is set to 'ObjectValueHelp'. The 'List Entries on Load' checkbox is checked. The 'Number of Lines' is set to 5. The 'Task Defaults' section includes 'Read Only' and 'Mandatory' checkboxes, both of which are unchecked.

Property	Value
Display Name	#MX_MX_SALUTATION_DN
Tooltip Text	
Presentation	ObjectValueHelp
Confirm Input	☐
Hide Input	☐
Show Search Field	☐
List Entries on Load	☒
Number of Lines	5
Read Only	☐
Mandatory	☐

5. Choose the *Attribute values* tab page.
6. Enter **mx_i_AttrValueHelp** as the *Value help Table name* and set the language dependency option.
7. Set the *Values ID* to the attribute name with the following exceptions as shown in the table below. For these attributes, use the values ID shown.

Values IDs

Attribute	Values ID
MX_ACADEMIC_TITLE_1	MX_ACADEMIC_TITLE
MX_ACADEMIC_TITLE_2	MX_ACADEMIC_TITLE
MX_NAME_PREFIX_1	MX_NAME_PREFIX
MX_NAME_PREFIX_2	MX_NAME_PREFIX

See the example below for the MX_SALUTATION attribute.

General | Storage | Presentation | Validation | Event Processes | **Attribute Values** | Entry Types

Legal Attribute Values

☐ Value:

☐ SQL query:

☒ Value Help
Table Name:
Values ID:
☒ Language Dependent

Related Information

[Attributes that Support Value Help \[page 1311\]](#)

4.2.2.2 Running the Read Help Values Job

Prerequisites

- In the Identity Management Developer Studio, you have imported the ABAP connector package that contains the Read Help Values job.
- In the Identity Management Administration User Interface, you have defined a repository on a repository type in the ABAP connector package.

Context

You can run the Read Help Values job in the Identity Management Administration User Interface and in the Identity Management Developer Studio.

To run this job in the Identity Management Administration User Interface, proceed as follows:

Procedure

1. Log on to the Identity Management Administration User Interface, accessible from `http://<host>:<port>/idm/admin`.
2. Choose the *System Configuration* tab and then *Repositories* from the menu on the left.
3. Select a repository that is defined on a repository type of the ABAP connector package.
4. In the *Details of Repository* pane, choose *Jobs* tab and select *Read Help Values*.
5. Run the job.

To run the Read Help Values job in the Identity Management Developer Studio, you need to select the job and choose *Run with Repository* from the context menu. In the *Select Repository* dialog box, select a repository and choose *OK*.

Note

There are also passes for reading content for system-specific attributes such as MX_START_MENU or MX_PARAMETER. Although reading and provisioning such data is supported by the ABAP connector, these passes are deactivated by default because there is no support in the provisioning framework for storing and provisioning this data in a system-specific manner. If you want to read and provision this data, you must implement the provisioning rules yourself to meet your needs.

Related Information

[ABAP Connector Package \[page 438\]](#)

4.2.3 Setting Up an SAP HCM System

Prerequisites

- The Virtual Directory Server is installed.
- The Identity Center is installed and configured.
- The SAP HCM system is installed and contains employee data.
- The user who exports the data from the SAP HCM system has the following PFCG roles assigned:
 - SAP_HR_LDAP_EXTRACTION (execution of data extraction)
 - SAP_HR_LDAP_PREPARE_EXTRACTION (needed for attribute mapping)
- The user who provisions employee-related information from SAP Identity Management to SAP HCM needs write authorization, therefore has the SAP_HR_COMM_IDM_INTEGRATION role assigned.
- An LDAP Connector is configured on the SAP HCM system.
- You have decided how to assign a user account name to an employee. See the *Limitations and Considerations When Using the SAP HCM Use Case*.

Context

When connecting an SAP HCM system to SAP Identity Management, the identity data is exported from the SAP HCM system and imported into the Identity Center. For this, the Virtual Directory Server is used as the common interface for processing the data. You can therefore use the export functions in SAP HCM that are available for exporting data to an LDAP directory. This data is then imported into a staging area in the Identity Center before being replicated into the productive identity store. Once the data is in the productive identity store, it can be provisioned to the connected systems, for example, another LDAP directory server.

Using a staging area instead of writing directly to the identity store has the following advantages:

- You can work with the data in the staging area before processing it further. For example, you can also set up the Workflow approval tasks to access the data in the staging area before writing it to the productive identity store.
- If you make changes to the database schema used for identity data in the SAP HCM system, you can adjust the attribute mapping in the staging area accordingly and you do not have to change the productive identity store's schema.

Note

We provide a template that contains two different files for setting up the staging area in the Identity Center.

To set up an SAP HCM system, proceed as described below.

Procedure

1. Create a new identity store and name it HCM_Staging_Area.
2. Import the staging area schema and hcm package files.
3. Adjust the event handling for the MX_HCM_EMPLOYEE entry type.
4. Set up the Identity Center to assign the user account name.
5. Configure the Virtual Directory Server.
6. Configure the SAP HCM system and export the data.

In systems with SAP ERP 6.0 SAP Enhancement Package 5 or higher, you can use the LDAP wizard (transaction HRIDMWIZARD_START) instead of the manual procedure described in this step.

Related Information

[Restrictions When Using the SAP HCM Use Case \[page 1330\]](#)

4.2.3.1 Create an HCM Staging Area Identity Store

Context

To create an HCM Staging Area identity store, do the following:

Procedure

1. Select the *Identity Management database* node in the tree view and choose ► **New** ► *Identity Store* ► from the context menu.
2. In *Create New Identity Store* dialog box, enter **HCM_Staging_Area** for the name of the new identity store and provide the name of the Administrator.
The identity store is created with default values and the corresponding node is added to the tree view.
3. The default properties of the identity store are shown in the properties view.
4. Select a tab to enter information about the identity store:
 - *General*

- [Password policy](#)

Related Information

[General \(Identity Store\) \[page 972\]](#)

[Password Policy \(Identity Store\) \[page 973\]](#)

4.2.3.2 Importing the Staging Area Template

Prerequisites

- You have followed the procedure in section *Reading Value Help Content* for your HCM system. Otherwise you will get the following error message after running an HCM extract: **Value not legal for this attribute: Attribute: MX_SALUTATION** when storing attribute MX_SALUTATION=value.

Context

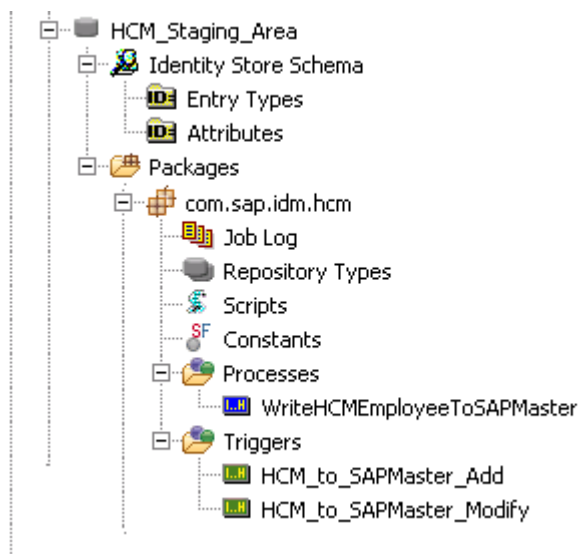
In this step, you import the template that contains the staging area schema to use for SAP HCM, as well as the corresponding process and action tasks for writing the identity data from the staging area to the productive identity store.

In the Identity Management Developer Studio, navigate to HCM_Staging_Area identity store.

Procedure

1. In the Identity Management Developer Studio, select the *Identity Store Schema* of the HCM_Staging_Area identity store and choose *Import* from the context menu.
2. Navigate to the HCM_Staging_Area.idmschema file which is located in <Install_folder>\ConfigurationPackages\HCM folder and choose *Open*.
3. In the HCM_Staging_Area identity store, select the *Packages* folder and choose *Import* from the context menu.
4. Navigate to the com.sap.idm.hcm.idmpck file of the HCM Integration Package which is located in <Install_folder>\ConfigurationPackages\HCM folder and choose *Open*.

The staging area MX_HCM_EMPLOYEE entry type and the corresponding process, triggers and action tasks are imported into the Identity Center. See the figure below.



Note the identity store ID for the HCM staging area identity store. You need it when configuring the Virtual Directory Server. Make sure your HCM Staging area identity store ID is filled with a value after the import. This value must not collide with other identity store IDs that are already present.

5. On the [General](#) tab of the new identity store, set the [Automatically create attributes](#) flag.

Also note the numeric ID for the SPML staging area identity store, which you can find on the same tab in the [ID/Name](#) field. Store the ID in the HR_STAGING_AREA_IDS_ID package constant under [Identity Center > Identity Store > Packages > com.sap.idm.hcm > Constants](#). By default the value is set to 2. In case you already have other identity stores, you need to change this value to the actual number. You need this value again when configuring the Virtual Directory Server.

Note

Make sure that both action tasks HCM Master Check for SYSUNAME of Employee and Write HCM Employee to SAP Master part of "WriteHCMEmployeeToSAPMaster" process are enabled after import.

Note

Note that during import, one or more Update global script dialog windows may appear. Select Use Existing and set the flag Use this action for all matching global scripts and choose Next.

Note

When importing `com.sap.idm.hcm.idmpck` file, make sure your dispatcher is selected to run the staging area's jobs.

4.2.3.3 Adjusting the Event Handling for MX_HCM_EMPLOYEE

Context

After importing the HCM staging area template files, adjust the event handling for the MX_HCM_EMPLOYEE entry type.

Procedure

1. Open [Identity store schema](#) > [Entry types](#).
2. Select the [MX_HCM_Employee](#) entry type with a double-click.
3. Choose the [Event Process](#) tab.
4. On the [Event Process](#) tab, select the [HCM_to_SAPMaster_Add](#) and [HCM_to_SAPMaster_Modify](#) respectively for the [Add Entry Process](#) and [Modify Entry Process](#). You can find these in the Select screen under [Available processes/forms](#). See the figure below.

The screenshot shows the 'Event Processes' tab of the configuration screen for the MX_HCM_EMPLOYEE entry type. The tab is selected, and the 'General' tab is also visible. The 'Relations' and 'Attributes' tabs are also visible. The 'Add Entry Process' field is set to '1167/HCM_to_SAPMaster_Add'. The 'Modify Entry Process' field is set to '1171/HCM_to_SAPMaster_Modify'. The 'Delete Entry Process' field is set to '-- None --'. Each field has a dropdown arrow button to its right.

Field	Value
Add Entry Process:	1167/HCM_to_SAPMaster_Add
Modify Entry Process:	1171/HCM_to_SAPMaster_Modify
Delete Entry Process:	-- None --

Note that there is no Delete Entry Process. This is due to the fact that users which are deleted in HCM are marked inactive in the Identity Management, using a Modify operation.

4.2.3.4 Setting up the Identity Center to Assign the User Account Name

Context

By default, for the SAP HCM use case, the user account name is determined by the SAP HCM system using the [P0105-SYHR_A_P0105_AF_SYSUNAME](#) field. As an alternative, the provisioning framework also supports using the [P0000-PERNR](#) field. To activate this mechanism, see the procedure below.

If you want to set up other mechanisms, then you must manually modify the tasks mentioned below according to your needs.

Procedure

1. In the Identity Management Developer Studio, navigate to the action task for *HCM Master Check for SYSUNAME of Employee*.
2. To have the Identity Center determine the user name account using the *P0000-PERNR* field, disable this task. For the default set-up (use the *SYSUNAME* field), leave the task as it is.
3. Navigate to the *Write HCM Employee To SAP Master* pass in the task with the same name.
4. Choose the *Destination* tab page.
5. Depending on your set up, make sure the MSKEYVALUE attribute has the following value.

Values for MSKEYVALUE

Set Up	Value for MSKEYVALUE
User account name is determined by the SAP HCM system using the <i>SYSUNAME</i> field (default).	<code>\$FUNCTION . sap_getSysUname(%P0000-PERNR%! !%P0105-SYHR_A_P0105_AF_SYSUNAME%! !%\$WRITE_FUTURE_DATED_HIRES%)\$\$</code>
User account name is determined by the Identity Center using the <i>PERNR</i> field.	<code>\$FUNCTION . sap_calcID(%P0000- PERNR%! !%P0105- SYHR_A_P0105_AF_SYSUNAME%! !%MSKEYVALUE%)\$\$</code>

6. Enable and disable the MSKEYVALUE rows accordingly and move the active MSKEYVALUE entry to the first row. It is then displayed in blue.

Note

This `sap_getSysUname` function could call a custom exit named `custom_generateHRID(Par )` that uses the attributes *P0000-PERNR*, *SYHR_A_P0105_AF_SYSUNAME* and *Employee-Key* (which is the MSKEYVALUE of the HCM staging area) as input parameters.

Currently, the `custom_generateHRID` function returns an empty string. If necessary, change this function to adjust the MSKEYVALUE to fit your needs.

→ Tip

You can find the custom exit jsript `custom_generateHRID` under ► *Packages* ► *com.sap.idm.hcm* ► *Scripts* .

4.2.3.5 Configuring the Virtual Directory Server

Prerequisites

- You have maintained the database connection for the identity store in Identity Center and know the password for the database user.
- The JDBC driver to use to access the Identity Center database is maintained in the class path for the Virtual Directory Server. (Maintain the driver under **Tools > Options > Classpath**.)
- You have imported the HCM staging area identity store into the Identity Center.

Context

In this section, we describe how to configure the Virtual Directory Server (VDS) so that the SAP HCM system can connect to it for the data export.

Procedure

1. Start the Virtual Directory Server console.
2. To maintain the configuration, choose **File > New**.
3. On the *New configuration* dialog that appears, choose **Group SAP NetWeaver 7.2 > Template HCM LDAP EXTRACT for IDM** and then choose **OK**.
4. Configure the parameters to use for the VDS as shown in the table below.

VDS Parameters

Field	Value	Example	Comment
Port	389	389	Select a different port if 389 is already being used.
Display Name	<Name_of_VDS>	Identity Store	
Identity Center URL	<Database_Connection_Parameters>	jdbc:sqlserver://localhost:1433; database=mxmc_db; user=mxmc_rt; password=<password>	Use the wizard (three dots) to maintain the URL. Examples for Microsoft SQL Server parameters are shown in the table that follows.

Field	Value	Example	Comment
Identity Store ID	<Staging_Area_Identity_Store_ID>	2	This ID was determined when setting up the staging area identity store.
Path to keys.ini	<Path_to_keys.ini_File>	C:\usr\sap\ldm\ Identity Center\ Key\Keys.ini	Make sure the path and file name use the syntax needed for your operating system.
Username	<Directory_Server_User>	hruser	This is the user that is used for the bind to the VDS. It is created if it does not yet exist.
Password	<Directory_Server_User_Password>	<password>	

VDS Parameters: Examples

Field	Value	Example	Comment
Server	<Server_Name>	localhost	
Port	<Port>	MSSQL: 1433 Oracle Thin Driver: 1521 DB2: 50000	
Database	<Database_Identifier>	mxmc_db	
User	<Database_User>	mxmc_rt	Runtime user
Password	<Database_User_Password>	<password>	You specified the password for the runtime user during the installation of the Identity Center database.
SID	<Service_Name>		Field only relevant for Oracle.
Schema	<Prefix_oper>	mxdb2_oper	Field only relevant for DB2.
FunctionPath	<Prefix_oper>	mxdb2_oper	Field only relevant for DB2.

5. Save the configuration.

Note

When configuring VDS for connecting SAP HCM with SAP Identity Management on Microsoft Windows, we recommend that you run VDS as a Windows service. This will ensure it is running even when the user is logged off.

Alternatively, you can run VDS in application mode, but in this case the application will not be running when the user is logged off.

For more information about the options you can run the server, see: [Configuring general parameters for a server](#).

6. Start the server.

4.2.3.6 Configure the SAP HCM System and Export the Data

Context

This topic describes how to configure the SAP HCM to export data to the Virtual Directory Server.

Procedure

1. Create the query to use for the export.
2. Maintain the attribute mapping between the HR fields and the input attributes used by the LDAP synchronization.
3. Create an RFC destination to use for the connection to the VDS.
4. Configure the parameters to use for this connection.
5. Maintain the mappings between the attributes used by the LDAP synchronization and the VDS.
6. Export the data.

→ Tip

Because the VDS does not use a specific LDAP schema for attributes, you can freely choose names for the attributes. To make maintenance easier, we recommend using the same attribute names throughout all of the mappings.

4.2.3.6.1 Creating the Query to Use for the Export

Context

In this step, you set up the query to use for the export. For this purpose, you can use the existing `LDAPEXTRACT604` query as a template. This query is assigned to the user group `SAPQUERY/L1`.

Create or modify the query in the SAP HCM Customizing development system and transport it to the productive system.

Procedure

1. Start query maintenance (transaction `SQ01`).
2. Choose **Edit** > *Other user group* and select the `SAPQUERY/L1` user group.

The queries available for this user group are displayed.

3. Select `LDAPEXTRACT604` with a double-click and choose **Query** > *Copy*.
4. In the dialog that follows, enter a name for the new query, for example, `LDAP_QUERY`.
5. Select this query with a double-click and choose **Query** > *Change*.

The attributes for the query appear.

6. To see the HR fields used by this query, choose *Basic List*.
7. To view the fields that are already selected in the example query, expand the data fields. These are the fields that are supported by the provisioning framework for SAP systems.

See the figure below for a subset of the fields for Infotype 0000.

Data fields	List field	Selection	Technical Name
HR InfoSet: Logical database	51	0	
HR Master Record: Infotype 0000 (Actions)	3	0	
HR Master Record: Infotype 0000 (Actions)	2	0	P0000
Personnel Number	☒	☐	P0000-PERNR
Action Type	☐	☐	P0000-MASSN
Reason for Action	☐	☐	P0000-MASSG
Customer-Specific Status	☐	☐	P0000-STAT1
Employment Status	☒	☐	P0000-STAT2
Special Payment Status	☐	☐	P0000-STAT3
Additional Fields	1	0	---
Text: Personnel Number	☐	☐	TEXT_P0000_PERNR
Text: Action Type	☐	☐	TEXT_P0000_MASSN
Text: Reason for Action	☐	☐	TEXT_P0000_MASSG
Text: Customer-Specific Status	☐	☐	TEXT_P0000_STAT1
Text: Employment Status	☒	☐	TEXT_P0000_STAT2
Text: Special Payment Status	☐	☐	TEXT_P0000_STAT3
Leaving date	☐	☐	SYHR_A_P0000_AF_FIREDATE
Entry Date	☐	☐	SYHR_A_P0000_AF_HIREDATE
Length of Employment Period (Number of years)	☐	☐	SYHR_A_P0000_AF_NOYRS

⚠ Caution

The attributes marked in the [LDAPEXTRACT604](#) sample query are supported by the SAP provisioning framework, meaning they are included in the schema and in the passes used for writing to the staging area and the productive identity store.

You can remove certain attributes from the query if you do not need to have them provisioned. However, if you add attributes, you will receive error messages when attempting to process the data further.

8. Note the Infotype number that the attributes belong to. You need this number when you maintain the attribute mapping in the next step.
9. Save the query and return to the initial screen.
10. Activate the query by executing it.

ℹ Note

You must execute the query once so that the query is generated and available for later steps.

→ Tip

You can reduce the amount of data selected for this initial execution of the query by using a limited time period (for example, [Today](#)) and a range for the [Personnel Number](#) (for example, 1 to 1000).

4.2.3.6.2 Specifying the Attribute Mapping Between the HR Fields and LDAP Synchronization

Prerequisites

The query used for extracting the data is active.

Context

In this step, you map the HR fields that are selected by the query to the input attributes used by LDAP synchronization.

Procedure

1. Using field assignment maintenance (transaction HRLDAP_MAP), select the [Global Work Area](#) indicator.

2. Enter **/SAPQUERY/L1** as the *User Group*.
3. Enter the name of your query, for example, **LDAP_QUERY**.
4. Choose *Import*.

The fields assigned to your query appear.

5. Maintain the *Attribute Grp* and *Attrib.Name* fields for each query field. Specify the attribute group so that it corresponds to the Infotype number you noted in the last step.

→ Tip

We recommend using the query field names as the attribute names.

ⓘ Note

To omit a field, set the *Tech. Field* field. Fields marked as such are not exported.

The table below shows a subset of the fields, based on the sample query used in *Creating the Query to Use for the Export*.

VDS Parameters: MS SQL Example

Query Fld	Description	Attribute Grp	Attrib.Name
P0000-PERNR	Personnel Number	P0000	P0000-PERNR
P8003-OBJID	Object ID	P8003	P8003-OBJID
SYHR_A_P0002_AF_SPLIT_BEG	Split Start	P0002	SYHR_A_P0002_AF_SPLIT_BEG
SYHR_A_P0002_AF_SPLIT_END	Split End	P0002	SYHR_A_P0002_AF_SPLIT_END
P0002-NACHN	Last Name	P0002	P0002-NACHN
P0002-VORNA	First Name	P0002	P0002-VORNA
P0002-NAME2	Name at Birth	P0002	P0002-NAME2
P0001-ENAME	Formatted Name of Employee or Applicant	P0001	P0001-ENAME
TEXT_P0002_TITEL	Text: Title	P0002	TEXT_P0002_TITEL
P0002-TITEL	Title	P0002	P0002-TITEL
P0002-ANRED	Form-of- Address Key	P0002	P0002-ANRED
TEXT_P0002_ANRED	Text:Form-of- Address Key	P0002	TEXT_P0002_ANRED
...	...	...	...

Query Fld	Description	Attribute Grp	Attrib.Name
SOURCE_SYSTEM	Source System	P0002	SOURCE_SYSTEM
SYHR_A_P0001_AF_PL_ST RAS	House Number and Street	P0001	SYHR_A_P0001_AF_PL_ST RAS
SYHR_A_P0001_AF_PL_RO OM1	Physical Room Number	P0001	SYHR_A_P0001_AF_PL_RO OM1
SYHR_A_P0001_AF_P_IS_ MGR	Personnel Number is Man- ager	P0001	SYHR_A_P0001_AF_P_IS_ MGR

6. Save the data.

Related Information

[Creating the Query to Use for the Export \[page 611\]](#)

4.2.3.6.3 Creating an RFC Destination to Use for the LDAP Connector

Context

Procedure

- Using destination maintenance (transaction SM59), create an RFC destination with the following properties:
 - Type:** T (TCP/IP Connection)
 - Name:** <Destination_Name> (for example, LDAP_VD)
 - Activation Type:** Registered server program
 - Program ID:** <Program_ID> (for example, LDAP_VD)

→ Tip

For easier reference in traces and logs, use the RFC destination name for the program ID.

- *Gateway host*: <Gateway_host> (host where the system's gateway runs)
- *Gateway service*: <Gateway_service> (name of the gateway service, for example sapgw<sys_nr>)
- *MDMP & Unicode* tab page under *Communication Type with Target System*: Select the *Non-Unicode* or *Unicode* radio button according to whether you extract Unicode data or not.

2. Save the data.

4.2.3.6.4 Configuring the Parameters to Use for the Connection to the VDS

Context

Procedure

1. Using directory service connection maintenance (transaction LDAP), set up the LDAP connector.
2. Set up a service user to use for the connection. Choose *System Users*, switch to edit mode, and choose *New Entries*.
3. Enter the properties for the system user.
4. Create an entry for the LDAP server.

4.2.3.6.4.1 Setting Up the LDAP Connector

Context

Procedure

1. Choose *LDAP Connectors*.
2. On the *LDAP Connector (Maintenance View)* screen that appears, choose *Display/Change* to switch to edit mode.

3. Choose [New Entries](#).
4. Enter the name of the RFC destination you created in the last step (for example, **LDAP_VD**).
5. Maintain the LDAP connector settings as necessary.

→ Tip

The application server name is the name of the server as specified in the `rdisp/myname` profile parameter. You can also check for the name to use using transaction **SM51**

6. Save the data.
7. To activate the connector, choose the [Start Connector](#) pushbutton.
8. Return to the main screen for the directory service connection maintenance.

4.2.3.6.4.2 Entering the User's Properties

Context

Procedure

1. For the [Distinguished Name](#), use the user ID that you specified for the VDS in *Configuring the Virtual Directory Server*, step 4.
 - [User ID](#): <User_ID> (for example, HR_USER)
 - [Distinguished Name](#): <Directory_Server_User> (for example, hruser)
 - [Auth. mechanism](#): Simple Bind
 - [Credential storage](#): Simple Memory
2. For the [Credentials](#), choose the symbol for [Change](#) to enter the directory server user's password. (This password must also match the password specified for the directory server user in step 4 referenced above.)

See the figure below.

The screenshot shows the 'User ID' field set to 'HR_USER'. Below it, the 'LDAP System User' section is expanded, showing the 'Distinguished Name' field set to 'hruser'. The 'Only read auth.' checkbox is unchecked. The 'Auth. mechanism' is set to 'Simple Bind' and 'Credential storage' is set to 'Simple Memory'. The 'Credentials' checkbox is checked, and there are icons for editing (pencil) and deleting (trash) next to it.

3. Save the data and return to the main screen for directory service connection maintenance.

Related Information

[Configuring the Virtual Directory Server \[page 608\]](#)

4.2.3.6.4.3 Creating an Entry for the LDAP Server

Context

Procedure

1. Choose [LDAP Servers](#).
2. Choose [Display/Change](#) to change to edit mode.
3. Choose [New Entries](#).
4. Enter the properties for the VDS as follows:
 - [Host name](#): <VDS_Host>
 - [Port number](#): <LDAP_Port> (for example, 1389)
 - [Product name](#): <blank>
 - [Protocol version](#): LDAP version 3
 - [LDAP Application](#): Employee
 - [Default](#): Inactive (unless the VDS should be the default LDAP server)
 - [Base entry](#): (for example, o=idstore)
The path is defined in the virtual tree of the virtual directory server.
 - [System Logon](#): <User_ID> (Use the user ID you specified in the last step, for example, HR_USER.)
 - [Read Anonymously](#): Inactive
5. Save the data and return to the main screen for directory service connection maintenance.

4.2.3.6.5 Maintain the Attribute Mappings

Context

Procedure

1. Using directory service connection maintenance (transaction LDAP), choose [LDAP Servers](#).
2. Select the LDAP server to maintain (for example, **LDAP_VD**) so that the row is marked.

Server Names				
ServerName	Host name	Appl.	Default	Base
LDAP_VD	vdshostexample.com	Employee	☒	o=idstore

3. If you are not in edit mode, then switch to edit mode.
4. In the left frame, select [Mapping](#) with a double-click.
5. In the [Mapping Overview](#) screen that appears, enter **sapIdentity** in the [ObjectClasses](#) list.
6. Maintain the mappings between the fields used by the LDAP synchronization and the VDS.

Note

The mappings are available on the installation CD (or in the installation package) in the subfolder located in ► **IDENTITY CENTER CORE 8.0** ► [AdditionalComponents](#) ►, in the HCM Ldap Mapping.xml file. You can upload the mappings using the XML Import function.

To maintain the mappings manually, see *Maintaining Mappings Manually*.

7. Go back and save the data.

Example

For an example of the LDAP attribute mappings, see the figure below.

Mappings									
St	Numb	Struct	Fld	Attrib.	F	I	E	R	E
	1	EMPLOYEE	KEY	cn	☒	☒	☒	☒	☐
	2	P0000	P0000-PERNR	P0000-PERNR	☐	☐	☒	☐	☐
	3	P0003	P0003-OBJID	P0003-OBJID	☐	☐	☒	☐	☐
	4	P0002	SYHR_A_P0002_AF_SPLIT_BEG	P0002-SYHR_A_P0002_AF_SPLIT_BEG	☐	☐	☒	☐	☐
	5	P0002	SYHR_A_P0002_AF_SPLIT_END	P0002-SYHR_A_P0002_AF_SPLIT_END	☐	☐	☒	☐	☐
	6	P0002	P0002-NACHN	P0002-NACHN	☐	☐	☒	☐	☐
	7	P0002	P0002-VORNA	P0002-VORNA	☐	☐	☒	☐	☐
	8	P0002	P0002-NAME2	P0002-NAME2	☐	☐	☒	☐	☐
	9	P0001	P0001-ENAME	P0001-ENAME	☐	☐	☒	☐	☐
	10	P0002	TEXT_P0002_TITEL	P0002-TEXT_P0002_TITEL	☐	☐	☒	☐	☐
	11	P0002	P0002-TITEL	P0002-TITEL	☐	☐	☒	☐	☐
	12	P0002	P0002-ANRED	P0002-ANRED	☐	☐	☒	☐	☐
	13	P0002	TEXT_P0002_ANRED	P0002-TEXT_P0002_ANRED	☐	☐	☒	☐	☐

Related Information

[Maintaining Mappings Manually \[page 619\]](#)

4.2.3.6.5.1 Maintaining Mappings Manually

Context

Procedure

1. Using directory service connection maintenance (transaction LDAP), choose [LDAP Servers](#).
2. Select the LDAP server to maintain (for example, LDAP_VD) so that the row is marked.

Server Names				
ServerName	Host name	Appl.	Default	Base
LDAP_VD	vdshostexample.com	Employee	☒	o=idstore

3. If you are not in edit mode, then switch to edit mode.
4. In the left frame, select [Mapping](#) with a double-click.

5. In the [Mapping Overview](#) screen that appears, enter **sapIdentity** in the [ObjectClasses](#) list.
6. Create an entry that maps the structure EMPLOYEE, field KEY, to the attribute cn. To create a new entry, choose ► [Edit](#) ► [Add New Mapping](#) ►.
7. Specify the rest of the entries to map the fields used by the HR LDAP query to identically-named attributes.

⚠ Caution

The entries for [Structure](#) and [Field](#) must be identical to the [Attribute Grp](#) and [Attrib.Name](#) entries you created for the query mapping in *Specifying the Attribute Mapping Between the HR Fields and LDAP Synchronization*.

Mapping Data

Structure	Field	Attribute	Flags to Set
EMPLOYEE	KEY	cn	Filter Import Mapping Export Mapping RDN Mapping
P0000	P0000-PERNR	P0000-PERNR	Export Mapping
P8003	P8003-OBJID	P8003-OBJID	Export Mapping
P0002	SYHR_A_P0002_AF_SPLIT _BEG	SYHR_A_P0002_AF_SPLIT _BEG	Export Mapping
P0002	SYHR_A_P0002_AF_SPLIT _END	SYHR_A_P0002_AF_SPLIT _END	Export Mapping
P0002	P0002-NACHN	P0002-NACHN	Export Mapping
P0002	P0002-VORNA	P0002-VORNA	Export Mapping
P0002	P0002-NAME2	P0002-NAME2	Export Mapping
P0001	P0001-ENAME	P0001-ENAME	Export Mapping
P0002	TEXT_P0002_TITLE	TEXT_P0002_TITLE	Export Mapping
P0002	P0002-TITEL	P0002-TITEL	Export Mapping
P0002	P0002-ANRED	P0002-ANRED	Export Mapping
P0002	TEXT_P0002_ANRED	TEXT_P0002_ANRED	Export Mapping

8. Go back and save the data.

Related Information

[Specifying the Attribute Mapping Between the HR Fields and LDAP Synchronization \[page 612\]](#)

4.2.3.6.6 Exporting the Data

Context

To export the data, execute the `RPLDAP_EXTRACT_IDM` report. This report writes the HR data to the LDAP directory server, in this case, the VDS.

Note

The `HRLDAP_EXTRACT` report implements the `RPAD00INFTY` and `HRPAD00INFTYDB` BAdIs. Its implementation of the `UPDATE` method only marks the infotypes 0000, 0001, 0002, and 0105 as relevant for the LDAP extraction.

If you want to include other infotypes in the `UPDATE` method of the BAdI, create your own implementation of the `UPDATE` method for the infotype you want to extract.

For more information, see the documentation of `RPLDAP_EXTRACT_IDM` report (using, for example, transaction `SE38`).

Procedure

1. Using the ABAP editor (transaction `SA38`), enter **`RPLDAP_EXTRACT_IDM`** as the program and choose *Execute*.
2. Enter the criteria to use for the report. Note the following for the corresponding fields.
 - *RFC Connection*: <blank>
The system searches for an active connector.
 - *LDAP Server*: <LDAP_Server> (for example, `LDAP_VD`)
 - Data source: Enter the data that corresponds to the query you defined in *Creating the Query to Use for the Export*.

Note

For example:

- *Global Work Area*: Activate
- *User Group*: `/SAPQUERY/L1`
- *Name*: <Query_Name>, for example, `LDAP_QUERY`

- *Variant*: <blank>

- *Options*: Delete Person in Directory with Employment Status 3
Activate this option if you want to delete user master records for users who have left the company.

Note

The users are then deleted in the Identity Center's identity store, but not in the SAP HCM system.

3. Execute the report.

Related Information

[Creating the Query to Use for the Export \[page 611\]](#)

4.2.3.6.6.1 Scheduling the Job to Run Periodically

Context

To regularly export identity data, create a variant and schedule a job that runs using this variant. Proceed as follows:

Procedure

1. To create a variant, enter the data for the report and save it instead of executing it. When saving it, enter a name for the variant and a short description.
2. Using job maintenance (transaction SM36), create a job that executes the RPLDAP_EXTRACT_IDM report. Configure the frequency of the job execution under *Start condition* and configure the report and variant names under *Step*.

Results

After configuring the LDAP connection to the VDS and executing this report, the identity data is extracted from the SAP HCM system and written to the VDS and the staging area in the Identity Center.

In VDS, the identities are assigned to the MX_HCM_EMPLOYEE entry type. This allows the Identity Center to recognize events for the entry type and to trigger the corresponding tasks when the entry type is changed.

4.2.3.6.6.2 Troubleshooting the Data Export

You can troubleshoot problems with the export from the SAP HCM system with the following tools:

- View the log produced for the export from the SAP HCM system using `SPLDAP_DISPLAY_LOG_TABLES` report.
- View traces for the LDAP Connector using `RSLDAPTRACE` report.
- Check connections using `RSBDCOS0` report.
- Show the result using transaction `LDAP`.

4.2.4 Setting Up an AS ABAP System

The ABAP connector makes it possible for SAP Identity Management to provision ABAP users, user/role and user/profile assignments to and from Application Server ABAP.

Overview

The ABAP connector is used for communication between SAP Identity Management and AS ABAP systems. SAP Identity Management can provision ABAP users and role/profile assignments to and from AS ABAP by using `FromSAP` pass and `ToSAP` pass. The connector is delivered as a separate package in the Provisioning framework for SAP Identity Management 8.0. The name of the package is `com.sap.idm.connector.abap`.

To connect to an AS ABAP system and to provision users to SAP Identity Management, see *Connecting an AS ABAP System*.

Related Information

[Prerequisites \[page 623\]](#)

[Connecting AS ABAP System \[page 625\]](#)

[Determine the Repository Type of your ABAP Repository \[page 626\]](#)

4.2.4.1 Prerequisites

To connect to an AS ABAP system, make sure that the following prerequisites are present:

- SAP Identity Management 8.0 SP00 (or higher) is installed and licensed
- AS ABAP: Release 4.6 and higher
- The SAP Java Connector version 3 (JCo 3) is included in the installation of Runtime Components both for Microsoft Windows and Unix/Linux. See [Using the SAP Java Connector \(JCo\)](#)

- The user type for the technical user should be [System](#) user. For more information about how to create a technical user, see the procedure below.
- To make sure the technical user used for the ABAP connector only has the necessary authorizations in the back-end system, assign the `SAP_BC_SEC_IDM_COMMUNICATION` role to the user (see SAP Note [1557803](#)).

This role was updated with Release 7.0 SPS 2 with authorizations for using the CUA, and with Release 7.1 for authorizations to retrieve value help values. Therefore, if you are upgrading to SPS2 or Release 7.1 respectively, and want to use these features, then you must also upload the new version of the role, regenerate the corresponding profiles, and update the role assignment for the technical user.

- For enhanced Business Suite integration, the technical user also needs the `SAP_CA_BP_IDM_INTEGRATION` role.
- Automatic profile generation must be enabled on the AS ABAP so that changes to role assignments are automatically reflected in a user's profile. You can check this using table maintenance (for example, transaction SM30). Maintain the `PRGN_CUST` table. Make sure an entry with the name `AUTO_USERCOMPARE` exists in the table and that it contains the value YES.

→ Tip

If you do not activate `AUTO_USERCOMPARE`, then run the `PFCG_TIME_DEPENDENCY` report after executing any provisioning steps.

- Provisioning ABAP users with `MX_SECURITY_POLICY` attribute is supported for SAP NetWeaver AS ABAP with SAP Basis 7.31 and higher.
- You are familiar with the dependencies between packages. See [Package Dependencies \[page 404\]](#).
- You are familiar with the restrictions that apply to all connectors and to AS ABAP connector only. See [Restrictions that Apply to All Connectors \[page 1332\]](#) and [Restrictions for AS ABAP/SAP Business Suite System Connectors \[page 1333\]](#)

Creating a Technical User

To connect to an AS ABAP system, you need a technical user that is used to establish the connection. To create the technical user, you create a custom copy of the PFCG role `SAP_BC_SEC_IDM_COMMUNICATION` and rename it to `Z_SAP_BC_SEC_IDM_COMMUNICATION`.

1. In the [SAP Easy Access](#) menu, choose the [Create role](#) button or access transaction PFCG.
2. Enter the name of the delivered standard role **`SAP_BC_SEC_IDM_COMMUNICATION`**.
3. To copy the standard role, choose [Copy role](#) and enter a new name **`Z_SAP_BC_SEC_IDM_COMMUNICATION`**.

⚠ Caution

Do not change the delivered standard roles (`SAP_*`), but rather only the copies of these roles (`Z_*`). Otherwise, the standard roles that you have modified will be overwritten by newly delivered standard roles during a later upgrade or release change.

4. Save and generate the profile for the role.
5. To create the technical user, access transaction SU01 and assign the new custom role `Z_SAP_BC_SEC_IDM_COMMUNICATION`. You should select user type [System](#) for this user to exclude it from password change intervals.

4.2.4.2 Connecting AS ABAP System

Context

To connect to an AS ABAP system, proceed as follows:

Procedure

1. Determine which repository type you want to use. See [Determine the Repository Type of your ABAP Repository \[page 626\]](#).
2. Import the connector package that contains the repository type you want to use. See [Importing the Provisioning Framework for SAP Identity Management 8.0 \[page 487\]](#).
3. Create a repository based on this repository type and set the correct values for the repository constants. See [Creating Repositories \[page 493\]](#).

Repository Type	Repository Constants
<i>ABAPSpecificApplicationServer</i>	See Repository Constants for AS ABAP (Specific Application Server) [page 1238] .
<i>ABAPLoadBalancedConnection</i>	See Repository Constants for AS ABAP (Load Balanced Connection) [page 1235] .
<i>DualStackSpecificApplicationServer</i>	See Repository Constants for a Dual-Stack System (Specific Application Server) [page 1248] .
<i>DualStackLoadBalancedConnection</i>	See Repository Constants for a Dual-Stack System (Load Balanced Connection) [page 1244] .
<i>ABAPBusinessSuite</i>	See Repository Constants for Business Suite AS ABAP [page 1251] .

4. Run the Read Help Values job. See [Reading Value Help Content \[page 598\]](#).
5. Prepare the Initial Load job.
 - a. Determining the leading system for attributes. See [Determining the Leading System for Attributes \[page 663\]](#).
 - b. Determine how initial passwords are generated. See [Determine How Initial Passwords are Generated \[page 664\]](#).
 - c. Run a test initial load. See [Running a Test Initial Load \[page 665\]](#).
 - d. Fix any inconsistencies or other problems, such as inconsistencies with privileges, timeout due to large number of identities from an AS ABAP system and so on. See [Fix Inconsistencies or Problems \[page 665\]](#).

6. Run the Initial Load job. See [Running the Initial Loads \[page 667\]](#).

Once you have run the Initial Load job, the connector provisions the users to SAP Identity Management according to the mapping between Identity Management and AS ABAP attributes. See [Mapping Between Identity Management and AS ABAP Attributes \[page 1277\]](#) and [Additional Mapping Attributes for Enhanced SAP Business Suite Integration \[page 1289\]](#).

7. Clean up the collected data. See [Cleaning up the Collected Data \[page 668\]](#).

To secure the connection between the Identity Center and the ABAP system, you can use Secure Network Communications (SNC).

You can also configure the ABAP connector to provision productive passwords to AS ABAP systems.

Related Information

[Configuring the ABAP Connector to Use SNC \[page 1313\]](#)

[Provisioning Productive Instead of Initial Passwords \[page 661\]](#)

[Creating an Update Job for AS ABAP System \[page 669\]](#)

4.2.4.3 Determine the Repository Type of your ABAP Repository

To connect to an AS ABAP system, you need to create a repository based on a specific repository type. Determining which repository type to use depends on whether you want to connect to a single stack AS ABAP system or dual stack (AS ABAP and AS Java) system and whether you want to connect to one particular Application Server (AS) node or to a group of Application Server nodes.

In addition, you can connect to an AS ABAP system by using the BusinessSuite connector that contains its own repository type.

Note

If you want to change the repository type of a given repository, for example from `ABAPSpecificApplicationServer` to `ABAPLoadBalancedConnection`, you can do this in SAP Identity Management Administration User Interface. See [Changing Repository Type of Repositories \[page 499\]](#).

See your options in the tables below:

Single Stack AS ABAP System

Application Server Node		
Connections	Repository Type	Comments
Connect to one particular AS node	<code>ABAPSpecificApplicationServer</code>	This repository type is delivered with the ABAP connector package <code>com.sap.idm.connector.abap</code> .

Application Server Node Connections	Repository Type	Comments
Connect to a group of AS nodes for load balancing	ABAPLoadBalancedConnection	This repository type is delivered with the ABAP connector package <code>com.sap.idm.connector.abap</code> .
Dual Stack (AS ABAP and AS Java) System		
Application Server Node Connections	Repository Type	Comments
Connect to one particular AS node	ABAPSpecificApplicationServer	This repository type is delivered with the ABAP connector package <code>com.sap.idm.connector.abap</code> .
	DualStackSpecificApplicationServer	This repository type is delivered with the Dual Stack connector package <code>com.sap.idm.connector.dualstack</code> .
<i>Note</i> The Dual Stack connector package depends on the ABAP connector package. You need to import the ABAP connector package first.		
Connect to a group of AS nodes for load balancing	ABAPLoadBalancedConnection	This repository type is delivered with the ABAP connector package <code>com.sap.idm.connector.abap</code> .
	DualStackLoadBalancedConnection	This repository type is delivered with the Dual Stack connector package <code>com.sap.idm.connector.dualstack</code> .
<i>Note</i> The Dual Stack connector package depends on the ABAP connector package. You need to import the ABAP connector package first.		

Application Server Node Connections	Repository Type	Comments
Connect to one particular AS node and to a group of AS nodes for load balancing	ABAPBusinessSuite	This repository type is delivered with the ABAP BusinessSuite connector package <code>com.sap.idm.connector.abap.businesssuite</code>. Use the ABAP BusinessSuite connector for integration with Business Suite applications and for delta handling, for example for role assignments. Note The delta handling is not enabled by default. For more information about delta handling for role/profile assignments, see the SAP Note 1626816. Note The ABAP BusinessSuite connector package depends on the ABAP connector package. You need to import the ABAP connector package first.

Once you choose the most appropriate repository type to use, you can proceed with importing the connector package that contains this repository type.

Related Information

[Setting Up an AS ABAP System \[page 623\]](#)

4.2.5 Setting Up an AS Java System

The Application Server (AS) Java connector makes it possible for SAP Identity Management to provision UME users, UME groups and user/role assignments to and from AS Java.

Overview

You can use the AS Java connector for communication between SAP Identity Management and AS Java systems. SAP Identity Management can provision UME users, UME groups and user/role assignments to and from AS Java using *FromSPML* pass and *ToSPML* pass. The connector is delivered as a separate

package in the Provisioning framework for SAP Identity Management 8.0. The name of the package is [com.sap.idm.connector.asjava](#).

The AS Java connector supports the following use cases:

- Connecting an AS Java system with database as UME data source
- Connecting an AS Java system with LDAP directory as UME data source

For more information how to configure the connector for your use case, see the Related Information section.

Related Information

[Prerequisites \[page 629\]](#)

[Connecting AS Java System with Database as UME Data Source \[page 630\]](#)

[Connecting AS Java System with LDAP Directory as UME Data Source \[page 631\]](#)

4.2.5.1 Prerequisites

To connect to an AS Java system, make sure that the following prerequisites are present:

Prerequisites

- SAP Identity Management 8.0 SP00 (or higher) is installed and licensed
- AS Java / J2EE Engine Release 6.40 and higher
- You need a technical user to establish the connection with an AS Java system. This technical user should only have the necessary authorizations in the back-end system. As of AS Java Release 6.40, the appropriate authorizations are provided with the UME action `SpmL_Write_Action`. There is also an action called `SpmL_Read_Action` for read-only access. Prior to Release 6.40, the action to use is `MANAGE_ALL_COMPANIES`.
- You are familiar with the restrictions that apply to all connectors and to AS Java connector only. See [Restrictions that Apply to All Connectors \[page 1332\]](#) and [Restrictions for AS Java System Connectors \[page 1334\]](#).

4.2.5.2 Connecting AS Java System with Database as UME Data Source

You can configure the AS Java connector to provision UME users, UME groups and user/role assignments to and from AS Java system with database as UME data source. In this way, the identity data is stored in the database of the AS Java.

Context

To connect to an AS Java System with database as UME data source, proceed as follows:

Procedure

1. Import the AS Java connector package. See [Importing the Provisioning Framework for SAP Identity Management 8.0 \[page 487\]](#).
2. Create a repository of type [ASJavaDB](#) with the correct values for the repository constants. See [Creating Repositories \[page 493\]](#) and [Repository Constants for AS Java \[page 1243\]](#).
3. Prepare the Initial Load job. This includes the following:
 - a. Determine the leading system for attributes. See [Determining the Leading System for Attributes \[page 663\]](#).
 - b. Determine how initial passwords are generated. See [Determine How Initial Passwords are Generated \[page 664\]](#).
 - c. Run a test initial load. See [Running a Test Initial Load \[page 665\]](#).
 - d. Fix any inconsistencies or other problems, such as out of memory error, timeout due to reading a large number of identities and so on. See [Fix Inconsistencies or Problems \[page 665\]](#).
4. Run the Initial Load job. See [Running the Initial Loads \[page 667\]](#).

Once you have run the Initial Load job, the connector loads the users to SAP Identity Management according to the mapping between Identity Management and AS Java attributes. See [Mapping Between Identity Management and AS Java Attributes \[page 1293\]](#).

5. Clean up the collected data. See [Cleaning up the Collected Data \[page 668\]](#).

You can also configure the AS Java connector to provision productive passwords to AS Java systems.

Related Information

[Provisioning Productive Instead of Initial Passwords \[page 661\]](#)

4.2.5.3 Connecting AS Java System with LDAP Directory as UME Data Source

You can configure the AS Java connector to provision UME users, UME groups and user/role assignments to and from AS Java system with LDAP directory as UME data source. In this way, the users, groups and group assignments are stored and managed in the LDAP directory while the Java privilege assignments are stored in the database of the AS Java.

Context

To connect to an AS Java system with LDAP directory as UME data source, proceed as follows:

Procedure

1. Import the AS Java connector package. See [Importing the Provisioning Framework for SAP Identity Management 8.0 \[page 487\]](#).
2. Create a repository that represents your LDAP system. For example, create a repository of type *ActiveDirectory* and set the correct values for the repository constants. See [Creating Repositories \[page 493\]](#) and [Repository Constants for Active Directory for Provisioning Framework \[page 1256\]](#).
3. Create a repository of type *ASJavaLDAP* and set the `BACKEND_REPOSITORYNAME` repository constant to be the name of the repository of type *ActiveDirectory*. See [Creating Repositories \[page 493\]](#) and [Repository Constants for AS Java \(LDAP Backend\) \[page 1241\]](#).
4. Prepare the Initial Load jobs. This includes the following:
 - a. Determine the leading system for attributes. See [Determining the Leading System for Attributes \[page 663\]](#).
 - b. Determine how initial passwords are generated. See [Determine How Initial Passwords are Generated \[page 664\]](#).
 - c. Run a test initial load. See [Running a Test Initial Load \[page 665\]](#).
 - d. Fix any inconsistencies or other problems, such as out of memory error, timeout due to reading a large number of identities and so on. See [Fix Inconsistencies or Problems \[page 665\]](#).
5. Run the Initial Load jobs of the two repositories. See [Running the Initial Loads \[page 667\]](#).

Once you have run the Initial Load jobs, the connector loads the users to SAP Identity Management according to the mapping between Identity Management and AS Java attributes, as well as between Identity Management and Active Directory attributes. See [Mapping Between Identity Management and AS Java Attributes \[page 1293\]](#) and [Mapping Between Identity Management and Active Directory Attributes \[page 1304\]](#).

6. Clean up the collected data. See [Cleaning up the Collected Data \[page 668\]](#)

You can also configure the AS Java connector to provision productive passwords to AS Java systems.

Related Information

[Provisioning Productive Instead of Initial Passwords \[page 661\]](#)

4.2.6 Setting Up an SAP HANA System

This document helps you understand the main functionality of the SAP HANA connector and explains how to configure your Identity Management and SAP HANA systems so that the connector works properly.

Prerequisites

- You have added JDBC driver in Identity Management dispatcher classpath.
To do this, run the Dispatcher Utility and navigate to ► **Tools** ► **Settings** ► **DSE Class Path** ►. Set the **DSE Class Path** to be the full path to the JDBC driver, including the driver file itself. Then navigate to Identity Center home ./Service-Scripts, open `<dispatcher_name>.prop` file and verify the JDBC driver file is added in section DSECLASSPATH.
See [Defining the Settings for the Identity Management Dispatcher Utility](#) and [2355326](#) .
- You have a technical user on the SAP HANA system with the required permissions to create new technical users on the system. See [Prerequisites for SAP HANA Technical User \[page 636\]](#).

→ Remember

The technical user must not be deleted, because all assignments made using it would also be deleted. For more information about creating users on an SAP HANA system, see the SAP HANA Security Guide on the [SAP HANA Platform page](#).

- You are familiar with the restrictions that apply to all connectors and to SAP HANA connector only. See [Restrictions that Apply to All Connectors \[page 1332\]](#) and [Restrictions for SAP HANA Connectors \[page 1335\]](#).

Overview

The SAP HANA connector for SAP Identity Management is used for communication between SAP Identity Management and SAP HANA systems. The connector is delivered as a separate package in the Provisioning framework for SAP Identity Management 8.0. The name of the package is `com.sap.idm.connector.hana`. The connector supports the following operations:

- Loading standard and restricted database users
- Provisioning and de-provisioning of users with attributes
- Assigning roles to users or removing roles from users
- Assigning system, application or analytic privileges to users or removing privileges from users
- Modification of user mappings related to multiple SAML providers

To see the list of all defined SAML providers, log on to SAP Identity Management User Interface, select a person entry and choose ► [Change Identity](#) ► [Account Attributes](#) ► [SAML Mapping](#) ►. You can select a provider to form a mapping to an external identity (unique user ID for HANA system).

Supported Database Users

The SAP HANA connector supports the following database users:

Database Users	Description
Standard Users	In SAP HANA, standard database users can create objects in their own schema and read data in system views.
Restricted Users	In SAP HANA, restricted database users initially have no privileges. Restricted users are intended for provisioning users who access SAP HANA through client applications and who are not intended to have full SQL access via an SQL console.

The SAP HANA connector supports loading and creating of restricted users as of SAP HANA Platform 1.0 SP08 and higher. Converting a restricted user into a standard user (or vice versa) is supported as of SAP HANA Platform 2.0 SP00.

For more information about SAP HANA database users, see [User Types](#) in *SAP HANA Security Guide*.

Supported Roles

The SAP HANA connector, working with the SAP HANA Platform SPS 04 and higher, supports more than the runtime roles.

Supported Operations According to SPS of SAP Identity Management and SAP HANA Platform

	SAP Identity Management 7.2 SPS 08	SAP Identity Management 7.2 SPS 09 or higher	SAP Identity Management 8.0 SPS 04 or higher
SAP HANA Platform SPS 04	Catalog roles	Catalog roles	Catalog roles
		System privileges	System privileges
		Application privileges	Application privileges
			Analytic privileges

	SAP Identity Management 7.2 SPS 08	SAP Identity Management 7.2 SPS 09 or higher	SAP Identity Management 8.0 SPS 04 or higher
SAP HANA Platform SPS 05	Catalog roles	Repository roles	Repository roles
		Catalog roles	Catalog roles
		System privileges	System privileges
		Application privileges	Application privileges
		SAML	SAML
			Analytic privileges
SAP HANA Platform SPS 06	Catalog roles	Repository roles	Repository roles
		Catalog roles	Catalog roles
		System privileges	System privileges
		Application privileges	Application privileges
		SAML	SAML
		Validity	Validity
		X.509	X.509
			Analytic privileges
SAP HANA Platform SPS 07	Catalog roles	Repository roles	Repository roles
		Catalog roles	Catalog roles
		System privileges	System privileges
		Application privileges	Application privileges
		SAML	SAML
		Validity	Validity
		X.509	X.509
		SAP Logon Ticket	SAP Logon Ticket
		SAP Assertion Ticket	SAP Assertion Ticket
			Analytic privileges

For more information about SAP HANA database roles and the difference between Catalog roles and Repository roles, see [Database Roles](#) and [Catalog Roles and Design-Time Roles Compared](#).

Privileges

Since multiple assignments of privileges are not supported in the SAP HANA system, by default no grouping is set for the assignment grouping of privileges in the Identity Center. For this reason, grouping is disabled in existing SAP HANA repositories.

The system supports the following new privilege types:

Privilege Types	Description
SYSTEMPRIVILEGE	The new entry for the Identity Center is created with the following naming convention: PRIV:SYSTEMPRIVILEGE:<repository>:<HANA privilege>.
APPLICATIONPRIVILEGE	The new entry for the Identity Center is created with the following naming convention: PRIV:APPLICATIONPRIVILEGE:<repository>:<HANA privilege>.
ANALYTICPRIVILEGE	The new entry for the Identity Center is created with the following naming convention: PRIV:ANALYTICPRIVILEGE:<repository>:<HANA privilege>.

Authentication Methods

The SAP HANA connector supports the provisioning of the following authentication methods:

- Basic

Note

When you use the Basic authentication method, [Enable Password Provisioning](#) on the [General](#) tab of the identity store must be selected.

- Kerberos

Note

Be aware that if you disable Kerberos authentication for a given user in SAP Identity Management User Interface, the value of the Kerberos ID (external identity of the user) will be kept. If you enable it again, the new value of the Kerberos ID will overwrite the old one.

- SAML
- X.509
- SAP Logon Ticket
- SAP Assertion Ticket

On the user interface, these methods can be set or modified for any users, as long as the methods are also defined in the SAP HANA system.

Note

Make sure that at least one of the supported authentication methods is specified, otherwise creating SAP HANA user in the backend system will fail.

Connecting SAP HANA System

To connect to an SAP HANA system, follow the steps below:

1. Import the HANA connector package. See [Importing the Provisioning Framework for SAP Identity Management 8.0 \[page 487\]](#).
2. Create a repository of type *SAPHANA* with the correct values for the repository constants. See [Creating Repositories \[page 493\]](#) and [Repository Constants for SAP HANA Database \[page 1259\]](#).
3. Run the Initial Load job. See [Running the Initial Loads \[page 667\]](#).
Once you have run the Initial Load job, the connector loads the users to SAP Identity Management according to the mapping between Identity Management and HANA attributes. See [Mapping Between Identity Management and HANA Attributes \[page 1295\]](#).

For more information about securing the SAP HANA connection, see ► [SAP HANA Network and Communication Security](#) ► [Securing Data Communication](#) ► [Configuring SSL for ODBC/JDBC Client Access](#) ► in the *SAP HANA Security Guide*.

Related Information

[SAP HANA Security Guide](#)

4.2.6.1 Prerequisites for SAP HANA Technical User

To use SAP HANA connector for SAP Identity Management, you have to fulfill the following requirements:

- You have to create a technical user in SAP HANA with the following roles or privileges:
 - MONITORING role
 - ROLE ADMIN system privilege
 - USER ADMIN system privilege
 - GRANT_ACTIVATED_ROLE object privilege and set it for execution
 - REVOKE_ACTIVATED_ROLE object privilege and set it for execution
 - GRANT_APPLICATION_PRIVILEGE object privilege and set it for execution
 - REVOKE_APPLICATION_PRIVILEGE object privilege and set it for execution
 - GRANT_ACTIVATED_ANALYTICAL_PRIVILEGE object privilege and set it for execution
 - REVOKE_ACTIVATED_ANALYTICAL_PRIVILEGE object privilege and set it for execution
 - To grant privileges to other users and roles, a user must have the permission required to grant these privileges.

→ Recommendation

To use a permanent password that you do not have to reset, disable the password lifetime.

❖ Example

Creating a technical user JOHN with SQL statements:

```
CREATE USER JOHN PASSWORD Abc123;  
ALTER USER JOHN DISABLE PASSWORD LIFETIME;  
GRANT MONITORING TO JOHN;  
GRANT ROLE ADMIN TO JOHN;  
GRANT USER ADMIN TO JOHN;  
GRANT EXECUTE ON GRANT_ACTIVATED_ROLE TO JOHN;  
GRANT EXECUTE ON REVOKE_ACTIVATED_ROLE TO JOHN;  
GRANT EXECUTE ON GRANT_APPLICATION_PRIVILEGE TO JOHN;  
GRANT EXECUTE ON REVOKE_APPLICATION_PRIVILEGE TO JOHN;  
GRANT EXECUTE ON GRANT_ACTIVATED_ANALYTICAL_PRIVILEGE TO JOHN;  
GRANT EXECUTE ON REVOKE_ACTIVATED_ANALYTICAL_PRIVILEGE TO JOHN;
```

- Follow all other requirements for creating technical users in SAP HANA with SQL statements or via the Developer Studio.
- You have to re-import the provisioning framework.
- If you want the connector to support activated SAP HANA roles, system, application and analytic privileges, as well as the e-mail address, locale, time zone attributes and the custom user attributes, you have to create and run a new load job for your repositories. If the initial load job is already created, you just need to re-run it.

Related Information

[Creating a Technical User with SQL Statements](#)

Creating a Technical User via Developer Studio: ► [Security Administration](#) ► [Managing SAP HANA Users](#) ► [User Provisioning](#) ► [Creating Users](#) ►

[SAP HANA Security Guide](#)

[Determining the Leading System for Attributes \[page 663\]](#)

4.2.6.2 Setting Up and Provisioning Session Client and Custom User Parameters for SAP HANA Users

Context

You can set user parameters, such as session client or custom ones, for a user in SAP Identity Management and provision them to an SAP HANA system.

Procedure

1. Log on to SAP Identity Management Administration User Interface and select ► [System Configuration](#) ► [Repositories](#) ► [<your SAP HANA repository>](#).
2. On the [Constants](#) tab, select the HANDLED_USER_PARAMS repository constant and add the value next to the default parameters. In this example, the user parameter **CLIENT**, which corresponds to Session Client attribute in SAP HANA, and the custom user parameter **ROOM NUMBER** are added. Note that all listed user parameters must be separated by comma.

Details of Repository "HANA":

Constants Jobs

Edit constants of repository "HANA"

Save Refresh

Name	Value	Description
HANDLED_USER_PARAMS	EMAIL ADDRESS, TIME ZONE, LOCALE, CLIENT, ROOM NUMBER	List of comma separated HANA user parameter names which will be set by the HANA connector

3. Save your configuration.
4. Log on to SAP Identity Management User Interface, select the [Manage](#) tab and create a new user or search for existing one.
5. Open the respective form ([Create Identity](#) or [Change Identity](#)) and select the [Settings](#) tab.
6. Go to the [User Parameters](#) section and add the values in the expected format. For example, **CLIENT==001** and **ROOM NUMBER==123**

Expected format is <key>==<value>.

User Parameters

Add... Remove

User Parameters

CLIENT==001

New Value

New Value: ROOM NUMBER==123

Add Cancel

7. Save your configuration.

Results

Both user parameters **CLIENT**, which corresponds to Session Client attribute in SAP HANA, and **ROOM NUMBER** are set for the user and provisioned to the SAP HANA system.

HDB - C5256031

HDB (TECH_USER)

User User Parameters

C5256031

☐ Disable ODBC/JDBC access

Authentication

☐ Password
 Password*: Confirm*:

☐ Kerberos
 External ID*:

☐ SAML [Configure](#)
☒ SAP Logon Ticket

☐ X509 [Configure](#)
☒ SAP Assertion Ticket

Valid From:

Valid Until:

Session Client:

HDB - C5256031

HDB (TECH_USER)

User User Parameters

User Parameters

+

×

Parameter	Value
ROOM NUMBER	123

Related Information

[Repository Constants for SAP HANA Database \[page 1259\]](#)

4.2.7 Setting Up a SuccessFactors System

This document is provided to help you understand the principles of the SuccessFactors connector and to help you configure SuccessFactors and SAP Identity Management systems, so that the connector works properly.

Prerequisites

- When SuccessFactors is used as an employee master data system, Employee Central is active on SuccessFactors, and Employee Central SOAP API is enabled. This is necessary for the provisioning of the employees' data from SuccessFactors. For more information about enabling Employee Central, see *Employee Central Master Implementation Guide* on SAP Help Portal at: [SuccessFactors HCM Suite](#).
- The SuccessFactors OData API is enabled.
- Role-based permission is enabled.
- You have a technical user on SuccessFactors with the necessary permissions. See [Prerequisites for SuccessFactors Technical User \[page 642\]](#).
- You are familiar with the restrictions that apply to all connectors and to SuccessFactors connector only. See [Restrictions that Apply to All Connectors \[page 1332\]](#) and [Restrictions for SuccessFactors Connector \[page 1336\]](#).

Overview

The SuccessFactors connector is used for communication between SAP Identity Management and SuccessFactors systems. The connector allows SAP Identity Management to integrate SuccessFactors systems into centralized user management scenarios. SuccessFactors connector is delivered as a separate package in the Provisioning framework for SAP Identity Management 8.0. The name of the package is [com.sap.idm.connector.sfsf](#). The connector supports the following operations:

- Loading users (active and inactive) with attributes.
- Modifying users.

Note

You can modify the username and the email address of the user, that is, the MX_HCM_SYSUNAME and the MX_MAIL_PRIMARY attributes. Changes to other user attributes in SAP Identity Management are not provisioned to SuccessFactors.

- Generating custom username and email address.
- Assigning roles to users or removing roles from users.

For these operations, the connector uses a delta read mechanism which is necessary because of the large amounts of data in the SuccessFactors system. The mechanism is implemented in order to read the latest attribute value for any entity processed by the [Initial Load](#) job.

Authentication

The SuccessFactors connector supports two authentication methods:

- Basic authentication

Note

This authentication type is deprecated and will soon be out of maintenance. For more information, see [Deprecation of HTTP Basic Authentication for APIs](#).

- mTLS (mutual Transport Layer Security) - Certificate-based authentication using mutual Transport Layer Security (mTLS) that provides a more secure authentication option to its users.

Connecting SuccessFactors System

To connect to a SuccessFactors system, follow the steps below:

1. Import the SuccessFactors connector package. See [Importing the Provisioning Framework for SAP Identity Management 8.0 \[page 487\]](#).
2. In case **mTLS authentication** is used for communication between SAP Identity Management and SuccessFactors system, execute the steps below:
 1. Generate a certificate via one of the two possible approaches:
 - via SAP for Me (recommended)
 1. Follow the procedure described in [3469904](#).
 2. After you have generated a certificate, extract the certificate trust chain from it. It will be needed in Step b of the procedure.
 - via SAP Cloud Identity Services- Identity Authentication service:
 1. Execute Step 6 from [Add System as Administrator](#).
As a result, you will automatically download the certificate as .p12 file.
 2. Copy the generated value from the [Insert as Text](#) field and save it with .cer, .pem, or .crt extension. It will be needed in Step b of the procedure.

Note

To be able to use a certificate generated by SAP Cloud Identity Services- Identity Authentication service, your version of SAP JVM should be 8.1.078 or later. For more information, see [SAP JVM 8.1 \(SUPPORT PACKAGES AND PATCHES\)](#).

2. Login to your SAP SuccessFactors system and follow the procedure described in [Upgrade to X.509 Certificate-Based Authentication for Incoming Calls](#).

Note

Make sure you select *SAP Identity Management* in the *Integration Name* field.

3. Create a repository of type *SFSF* with the correct values for the repository constants. See [Creating Repositories \[page 493\]](#) and [Repository Constants for SuccessFactors \[page 1261\]](#).
4. Run the *Initial Load* job. See [Running the Initial Loads \[page 667\]](#).

Once you have run the [Initial Load](#) job, the connector loads the users to SAP Identity Management according to the mapping between Identity Management and SuccessFactors attributes. See [Mapping Between Identity Management and SuccessFactors Attributes \[page 1300\]](#).

All the changed values of these attributes in SuccessFactors are updated for every user in SAP Identity Management. If any of these users do not exist in SAP Identity Management, the connector creates them.

Note

When running the [Initial Load](#) job in delta mode, it is only possible to load changes up to three months into the past. This is due to a restriction in the Employee Central Compound Employee API that does not allow the `last_modified_on` filter parameter to be older than three months. For more information, see: [Thresholds and Limits of the CompoundEmployee API](#).

Any data changes older than this period will not be read and the job will result in an error. To load data changes older than that period, run the [Reset Delta](#) job followed by the [Initial Load](#) job, so that a full load is performed.

The [Initial Load](#) job can generate a custom username (MSKEYVALUE) for an employee in SAP Identity Management and then provision it back to SuccessFactors. See [Generating Custom Username](#) in the Related Information section.

Related Information

[SuccessFactors Employee Central Integration \[page 22\]](#)

[Generating Custom Username \[page 643\]](#)

4.2.7.1 Prerequisites for SuccessFactors Technical User

You need to have a technical user on SuccessFactors with the following permissions:

Group	Permission
Manage User	• Import Employee Data • Add New User • Employee Export • Manage Users • Set User Status • Reset User Account • Reset User Passwords
General User Permission	• SFAPI User Login • User Login

Group	Permission
Manage Integration Tools	Allow Admin to Access OData API through Basic Authentication
Employee Central API	- Employee Central HRIS OData API (editable) - Employee Central Foundation OData API (editable) - Employee Central Foundation SOAP API - Employee Central HRIS SOAP API - Employee Central HRIS OData API (read-only) - Employee Central Foundation OData API (read-only) - Employee Central Compound Employee API (restricted access)

The target for the permissions requiring it should be [All/Everyone](#).

4.2.7.2 Generating Custom Username

The Initial Load job can generate a custom username (MSKEYVALUE) for an employee in SAP Identity Management and then provision it back to SuccessFactors.

The following process takes place:

1. The [Initial Load](#) reads data from the SF Employee Central. When an employee is created in SF Employee Central, it might get a default username.
2. To generate a custom username for an employee, the [Initial Load](#) first checks if this MX_PERSON entry already exists in SAP Identity Management.
 - If the MX_PERSON entry already exists, the [Initial Load](#) uses the MSKEYVALUE of the old MX_PERSON entry. It is checked if an MX_PERSON entry exists, which has attribute MX_FS_PERSONNEL_NUMBER equal to **person_id_external**.
 - If the MX_PERSON entry does not exist, the [Initial Load](#) invokes the [custom_generateID](#) script from the Custom package. You can implement that script in the Custom package to generate a custom username.

The [custom_generateID](#) script receives all data that is read from SF Employee Central via Compound Employee API. The structure is a map object with the following keys: **person_id_external**, **person_id**, **logon_user_id**, **logon_user_name**, **personal_information**.

Personal Information object from SF Compound Employee API is a complex structure with various attributes like first_name, last_name, birth_name and so on. In addition, it supports validity, thus there could be multiple complex values with different validity periods (represented as attributes start_date and end_date).

All that information is concatenated in one string and then used as a value in the map object passed to the script. The delimiter that is used is **!!** for concatenating personal information values with different validity, and **!** for the personal information attributes. For example:

```
birth_name=NEW_HIRE_BN!created_by=53967!
created_on_timestamp=2015-03-26T07:15:58.000Z!end_date=9999-12-31!
first_name=NEW_HIRE_FN!formal_name=NEW_HIRE_FN
NEW_HIRE_MN NEW_HIRE_LN !gender=M!is_overridden=false!
last_modified_by=53967!last_modified_on=2015-03-26T07:15:58.000Z!
```

**last_name=NEW_HIRE_LN!marital_status=S!middle_name=NEW_HIRE_MN!
nationality=DEU!native_preferred_lang=DE!preferred_name=NEW_HIRE_PN!
salutation=MR!start_date=2015-03-26!**

Additional information is also received from the Compound Employee API and part of the map object. You can get all data by printing the Par argument of the script, for example calling `uWarning(Par)`;

- If the `custom_generateID` script returns an empty string, the *Initial Load* checks the username of the employee in the SF Employee Central. The *Initial Load* loads the username with a default value.
 - If the username is not empty, the MSKEYVALUE is set to be the username.
 - If the username is empty, the MSKEYVALUE is set to be the value of the **person_id_external** (one of the identifiers for an employee in SFSF)
- If the script returns a value, the MSKEYVALUE is set to be the result of the `custom_generateID` script.

Here is a sample script to generate the username based on the **person_id_external** value:

Sample Code

```
function custom_generateID(Par) {  
    var personIdExternal = Par.get("person_id_external");  
    var username = "E" + personIdExternal;  
    return username;  
}
```

Note

When you generate a custom username, be careful not to generate an already existing MSKEYVALUE. In this case, you will trigger an update of the user, not create.

Now that we have generated the MSKEYVALUE, the process continues as follows:

3. The modify triggers are set on the system privilege.
4. The MX_HCM_SYSUNAME attribute is set to be the MSKEYVALUE.
5. The *Modify User* process is triggered for MX_PERSON entries which username has been modified.
6. The *Modify User* process triggers the update to become updates of the username in SFSF.

Note

The *Modify User* process can also update the email of an MX_PERSON entry. By default the *Initial Load* does not set an email address. But if an email address is generated or modified in SAP Identity Management, it will be provisioned to SFSF.

4.2.8 Setting up an Identity Authentication System

The Identity Authentication connector makes it possible for SAP Identity Management to provision users to and from SAP Cloud Identity Services - Identity Authentication, formerly known as SAP Cloud Identity (SCI).

Prerequisites

You have an Identity Authentication tenant. See [Initial Setup](#) section in the *SAP Cloud Identity Services - Identity Authentication* documentation.

You have a technical user to establish the connection with SAP Cloud Identity Services - Identity Authentication, shortly Identity Authentication service. For this, you must add a system as a tenant administrator and assign the **Manage Users** administrator role for the system. See [Add System as Administrator](#).

Overview

The Identity Authentication connector is a standard REST-based connector. It is used for communication between SAP Identity Management (an SAP on-premise solution) and SAP Cloud Identity Services - Identity Authentication.

SAP Cloud Identity Services - Identity Authentication provides authentication and single sign-on for applications running on SAP Business Technology Platform (SAP BTP). It is tenant-aware, meaning each customer has its own tenant that represents a single instance of the Identity Authentication service with a specific configuration and data separation. The Identity Authentication connector is also tenant-aware and is able to provision users to and from a specific customer tenant by using ToSCIM and FromSCIM passes.

The connector is delivered as a separate package in the Provisioning framework for SAP Identity Management 8.0. The name of the package is [com.sap.idm.connector.sci](#).

Note

Currently, only [Identity Authentication SCIM REST API](#) is supported. As stated in the documentation, this API is deprecated. If you would like to use the [Identity Directory SCIM REST API](#), please use the [hybrid integration with the Cloud Systems via the IPS Proxy](#). You can set which API version you would like to use with a property on the IPS Proxy system. For more information, see [Identity Authentication](#).

The connector supports the following use cases:

- Loading users and their attributes to and from Identity Authentication service.
- Creating, deleting, enabling, disabling and modifying users, as well as setting a productive password for a user.

Note

If you create a user with MX_DEPARTMENT attribute, make sure the department exists (that is, the attribute has a valid value. Otherwise, the [CreateUser](#) process will fail.

Note

If you create a user without setting a productive password, make sure you provide a valid email address for this user. In this case, the user will receive an account activation email.

- User UUID (universal unique identifier). For more information, see [User UUID](#)

Note

The connector does not support assigning roles and groups to users.

Connecting Identity Authentication Tenant

To connect to an Identity Authentication tenant, proceed as follows:

1. Import the Identity Authentication connector package. See
 - [Identity Authentication Connector Package \[page 459\]](#)
 - [Importing the Provisioning Framework for SAP Identity Management 8.0 \[page 487\]](#)
2. Create a repository of type *SCI* for your Identity Authentication tenant and set the correct values for the repository constants. See
 - [Creating Repositories \[page 493\]](#)
 - [Repository Constants for Identity Authentication System \[page 1267\]](#)
3. Run the Initial Load job.

Once you have run the Initial Load job, the connector loads the users to SAP Identity Management according to the mapping between SAP Identity Management and Identity Authentication service attributes. See [Mapping Between Identity Management and Identity Authentication Service Attributes \[page 1306\]](#).

4.2.9 Setting up a SCIM System

The SCIM 2.0 connector makes it possible for SAP Identity Management to provision users to and from SCIM systems.

Overview

The SCIM connector uses a standard SCIM 2.0 protocol. It is delivered as a separate package in the Provisioning framework for SAP Identity Management 8.0. The name of the package is [com.sap.idm.connector.scim](#). The SCIM connector supports the following operations:

- Loading users with attributes
- Creating, modifying, deleting, enabling and disabling users
- Reading groups, if supportable

Basic and OAuth authentication methods are supported for establishing the connection to a SCIM-based system. Note that in case of OAuth, SAP Identity Management is using basic authentication only the first time it authenticates with an authorization server. Basic authentication in both methods (basic and OAuth) is of type challenge-response.

You can use the SCIM connector in the following use cases:

- **General usage** - for communication between SAP Identity Management and various SCIM backend systems (SAP or non-SAP).
- **Hybrid integration with cloud systems** - for communication between SAP Identity Management and various cloud systems supported by SAP Cloud Identity Services - Identity Provisioning.

The procedure below describes the general usage of the SCIM connector. To use the connector for hybrid integration with cloud systems, see [Setting Up Hybrid Integration with Cloud System \[page 647\]](#)

Connecting SCIM 2.0 System

1. Import the SCIM connector package. See:
 - [SCIM Connector Package \[page 462\]](#)
 - [Importing the Provisioning Framework for SAP Identity Management 8.0 \[page 487\]](#)
2. Create a repository of type *SCIM* for your SCIM system and set the correct values for the repository constants. See the General Usage table in [Repository Constants for a SCIM System \[page 1269\]](#)

Note

As of version 8.0 SP06, due to a change in the data type of the AUTH_PASSWORD and the PROXY_PASSWORD repository constants, if you reimport the SCIM connector package, you need to enter the value for those constants again for every repository of type SCIM that you have created.

3. Run the initial load job.
Once you run the *Initial Load* job, the connector loads the users to SAP Identity Management according to the mapping between SAP Identity Management and the attributes of the SCIM system. See: [Mapping Between Identity Management and SCIM Attributes \[page 1307\]](#)

4.2.10 Setting Up Hybrid Integration with Cloud System

You can connect SAP Identity Management to various cloud systems that are supported by SAP Cloud Identity Services - Identity Provisioning to provision users and authorizations across hybrid environments.

Prerequisites

- You have enabled the Identity Provisioning service for your subaccounts using SAP Business Technology Platform cockpit. See [Access the Identity Provisioning \(Standalone\)](#)

- You have access to the proxy systems in the Identity Provisioning UI. By default, the [Proxy Systems](#) tile is not displayed in the UI, that is why you need to create an incident to the BC-IAM-IPS component and request this tile.
- You have imported the latest version of the SCIM connector package [com.sap.idm.connector.scim](#).

Context

The integration between SAP Identity Management and SAP Cloud Identity Services - Identity Provisioning allows you to manage users and their authorizations, as well as groups (if supportable) across hybrid environments. To achieve this, SAP Identity Management uses the SCIM 2.0 connector.

For its part, the Identity Provisioning acts as a proxy between SAP Identity Management and the cloud system. The proxy reads the user data from the cloud system (based on the defined mapping between the cloud system entities and the data formats of the Identity Provisioning) and sends it to SAP Identity Management.

The user data from the cloud system is stored in SAP Identity Management. You can create new users, modify, delete or disable existing ones and provision your changes to the cloud system, if these operations are supported by the cloud system. For more information about the supported proxy cloud systems and operations, see [Proxy Systems](#)

Setting up the integration with a cloud system involves several steps in the Identity Provisioning service and the Identity Management in the exact order provided below.

SAP Cloud Identity Services - Identity Provisioning

Procedure

1. Configure the Identity Provisioning service for hybrid integration with SAP Identity Management. See [Hybrid Scenario: SAP Identity Management](#).
2. Export the newly created proxy cloud system as .csv file.

SAP Identity Management

Procedure

1. In SAP Identity Management Administration User Interface, select the [System Configuration](#) tab and import the .csv file.
A new repository of type SCIM is imported.
2. Select the [Constants](#) tab of the SCIM repository and enter the value for AUTH_USER and AUTH_PASSWORD constants. The values of all other constants are automatically filled in after the import. For more information, see the Identity Provisioning Service table with repository constants in [Repository Constants for a SCIM System \[page 1269\]](#)

Note

The generated *client ID* and the *Secret* (password) that you have provided during the registration of the OAuth client are used for configuring the AUTH_USER and AUTH_PASSWORD repository constants. These are the technical user credentials needed by SAP Identity Management to establish the connection with the backend system (in this case, the proxy system).

Note

As of 8.0 SP06, due to a data type change of AUTH_PASSWORD and PROXY_PASSWORD repository constants, if you reimport the SCIM connector package, you need to enter again their values for every repository of type SCIM you have created.

3. Select the *Jobs* tab and run the initial load job.

Once you run the *Initial Load* job, the connector loads the users to SAP Identity Management according to the mapping between SAP Identity Management and the attributes of the proxy system in the Identity Provisioning service. See: [Mapping Between Identity Management and SCIM Attributes \[page 1307\]](#)

Next Steps

The users and groups (if supportable) from the cloud system are loaded in the identity store of SAP Identity Management. You can proceed with creating new users, modifying, deleting or disabling existing ones and provision the changes to the cloud system.

→ Tip

If you need to delete or update an existing user in the SCIM system, you don't need to do the change there. Instead, make all changes in SAP Identity Management only.

Related Information

[System Types and Configurations](#)

[Hybrid Scenarios with Identity Provisioning Proxy](#) 

4.2.11 Setting Up an S/4HANA On-Premise System

The SAP S/4HANA connector is used for communication between SAP Identity Management and SAP S/4HANA (SAP Business Suite 4 SAP HANA) on-premise system.

Prerequisites

- You are running SAP Identity Management 8.0 SP08 or higher.
- You are running SAP S/4HANA on-premise system 1909 or higher.

Note

Running older SAP S/4HANA versions might also work, but this has not been thoroughly tested under all conditions. SAP therefore cannot guarantee or imply reliability or function of an integration scenario with older SAP S/4HANA versions.

- To provision User UUID (universal unique identifier) to S/4HANA on-premise, the following requirements should be fulfilled:
 - SAP Note [3003462](#) is applied on the S/4HANA system.
 - The communication (technical) user defined on the JCO_CLIENT_USER repository constant (ABAP or Business Suite repository) has the authorization object S_USER_UID assigned through a role.

Overview

The connector allows you to integrate SAP S/4HANA on-premise system into your SAP Identity Management landscape for managing and provisioning of business users.

A business user is a SU01 user that has a one-to-one relation to a corresponding business partner. In SAP S/4HANA, this is defined as a natural person who is represented by a business partner and a link to a user in the system. The business partner contains the personal information of the business user. This includes personal, work center, communication and organization data. For more information, see SAP Note [2570961](#).

The business partner itself can be a person, an organization (company, subsidiary) or a group of people or organizations in which your company has a business interest. The SAP S/4HANA connector supports business partners of category: person (natural person). Persons can have business partner roles assigned, such as: employee, freelancer and service provider.

Note

The SAP S/4HANA connector should always be used along with the ABAP or the BusinessSuite connector, as this will ensure data synchronization between SAP Identity Management, SAP S/4HANA and ABAP. No changes have been made in the ABAP connector and the Business Suite connector due to the implementation of the SAP S/4HANA connector.

Note

You cannot use the SAP S/4HANA connector for connecting SAP Identity Management with SAP S/4HANA cloud. This is only possible with SAP Cloud Identity Services - Identity Provisioning and its [Hybrid Scenario: SAP Identity Management \[page 647\]](#) where SAP S/4HANA cloud is added as a proxy system.

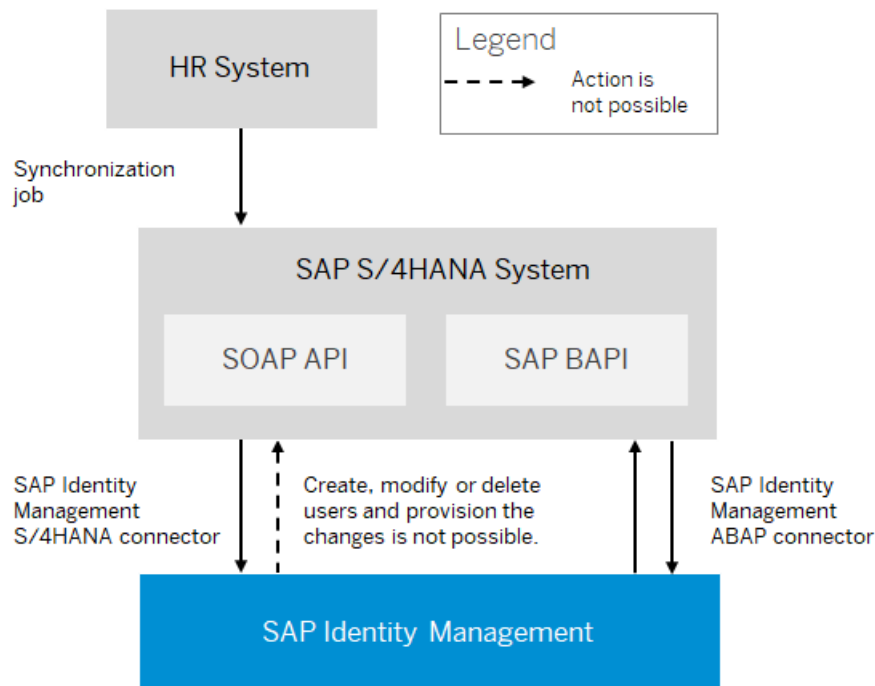
Provisioning of User UUID (generated by Identity Authentication service) from SAP Identity Management to SAP S/4HANA on-premise is supported by using the ABAP or the BusinessSuite connector.

The connector is delivered as a separate package [com.sap.idm.connector.s4hana](#) in the Provisioning framework for SAP Identity Management 8.0. For more information, see: [S4HANA Connector Package \[page 471\]](#)

Human Resource Integration Scenarios

There are two scenarios for using the SAP S/4HANA connector depending on whether the SAP S/4HANA integration with a human resource (HR) system, such as SAP HCM or SAP SuccessFactors, is active or inactive.

- **HR integration is active**

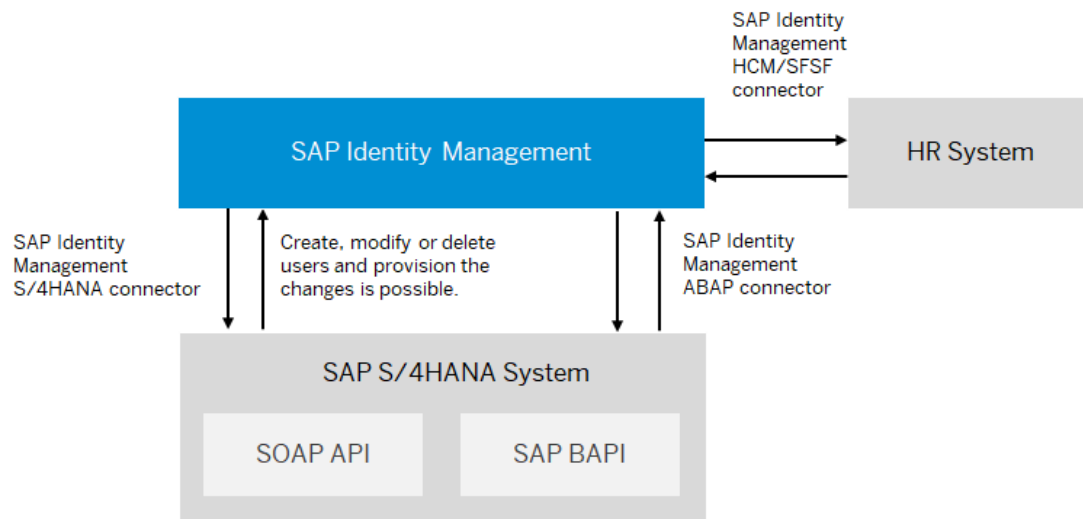


In this scenario, business users from the HR system are replicated to SAP S/4HANA. Synchronization of business users between both systems is managed with a scheduled sync report.

If HR integration is active, the SAP S/4HANA connector can only be used if you have run the ABAP or Business Suite initial load job and want to load business users with the new SAP S/4HANA attributes supported by SAP Identity Management. Creating, modifying or deleting business users and provisioning the changes to SAP S/4HANA is not possible. The same applies for business users which are not managed by an HR system (for example, business users created in SAP S/4HANA by SAP Fiori app). You cannot create those users in SAP Identity Management and provision them to SAP S/4HANA.

If HR integration is active, the IS_HR_INTEGRATION_ACTIVE repository constant must be enabled. If you then try to create and provision a business user to SAP S/4HANA, no request will be sent, and you will get a warning that the operation is not possible. If the repository constant is not enabled, and you try to create and provision a business user to SAP S/4HANA, you will get an error message informing you that this operation is not possible.

- **HR integration is inactive**



In this scenario, the HR integration switch in SAP S/4HANA is deactivated. The SAP S/4HANA connector can be used for loading and managing business users to and from an SAP S/4HANA system just like any other SAP Identity Management connector is used for managing users in the corresponding systems.

For more information, see *HCM Integration active or inactive: What does it mean?* in SAP Note: [2570961](#)

Connecting SAP S/4HANA System

Setting up the connection between SAP Identity Management and SAP S/4HANA on-premise involves the following high-level steps:

1. Import the SAP S/4HANA connector package.
See: [Importing the Provisioning Framework for SAP Identity Management 8.0 \[page 487\]](#)
2. Create a repository of type *S4HANA* and configure the repository constants.
See: [Repository Constants for SAP S/4HANA \[page 1274\]](#)
3. Run the ABAP or the Business Suite initial load job.

→ Recommendation

If you have an SAP S/4HANA system with existing business users, it is mandatory to run the ABAP or the Business Suite initial load job before the S/4HANA initial load. Otherwise, if SAP S/4HANA initial load is run first, this might cause an issue, where two separate person entries are created in SAP Identity Management - the SU01 user and its linked business partner, for one and the same business user in the SAP S/4HANA system. SAP Identity Management does not support cases where one business user is represented by two person entries. For more information, see: *Two Identities One Business Partner* section in [S/4HANA Initial Load Job \[page 654\]](#).

4. Read the value help of SAP S/4HANA attributes.

→ Recommendation

If you want to use the value help of certain SAP S/4HANA attributes, we recommend reading the value help before running the SAP S/4HANA initial load job. For more information, see: [S/4HANA Attributes Using Value Help \[page 653\]](#)

5. Run the SAP S/4HANA initial load job.

See: [S/4HANA Initial Load Job \[page 654\]](#)

Once you have run the Initial Load job, the connector loads the business users to SAP Identity Management in accordance with the mapping between SAP Identity Management and SAP S/4HANA attributes.

See: [Mapping Between Identity Management and SAP S/4HANA Attributes \[page 1308\]](#)

See: [S/4HANA Attributes](#).

4.2.11.1 S/4HANA Attributes Using Value Help

Before running the SAP S/4HANA initial load job, we recommend reading the value help of the SAP S/4HANA system attributes listed below. Otherwise, you may receive errors during the initial load.

- MX_S4HANA_FORM_OF_ADDRESS
- MX_ACADEMIC_TITLE_1
- MX_ACADEMIC_TITLE_2
- MX_S4HANA_LASTNAME_PREFIX
- MX_S4HANA_LASTNAME_SECOND_PREFIX
- MX_S4HANA_NAME_SUPPLEMENT
- MX_S4HANA_BUSINESS_PARTNER_ROLE_CODE

📘 Note

You can read the value help of SAP S/4HANA attributes manually. For more information how to do that, see [Defining Attribute Value Help \[page 183\]](#).

📘 Note

The value help content for MX_ACADEMIC_TITLE_1 and MX_ACADEMIC_TITLE_2 attributes is also provided in the AS ABAP systems. As it may differ from the value help in the SAP S/4HANA system, you need to decide which value help you want to read.

4.2.11.2 S/4HANA Initial Load Job

Prerequisites

- You have imported the [com.sap.idm.connector.s4hana](#) connector package.
- You have created a repository of type [S4HANA](#) and configured the repository constants.
- You have run the ABAP or the Business Suite initial load job.
- You have read the S/4HANA attributes using value help. See: [S/4HANA Attributes Using Value Help \[page 653\]](#)

Passes

The [Initial Load \(S4HANA\)](#) job contains passes that are common for all initial load jobs in SAP Identity Management connectors. Among them are the read and write S4HANA users passes:

Pass	Description
Read S4HANA users	From Custom pass using the new FromS4HANA.class Reads all business users from the given SAP S/4HANA repository based on the specific configuration and writes them into the <code>sap%\$rep.\$NAME%users</code> database table, created by the Identity Management runtime user.
Write S4HANA users	To Identity store pass Creates or updates the Identity Management users based on the data in the <code>sap%\$rep.\$NAME%users</code> database table. The attribute mapping between SAP Identity Management and SAP S/4HANA is defined on the Destination tab. All Identity Management attributes are regular attributes. Only the values of the SAP S/4HANA phone numbers and the business users' validity are formatted at runtime before being stored. See: S/4HANA Attributes.

MSKEYVALUE

The MSKEYVALUE of a newly created or updated user in SAP Identity Management is generated by the `sap_s4hana_generateMSKEYVALUE` javascript. The script takes the `PersonExternalID` and the `UserAssignment.UserID` of the business partner being processed, and makes the following checks:

- If `PersonExternalID` and `UserAssignment.UserID` are equal, it directly returns their value.
- If a user with `MSKEYVALUE = PersonExternalID` exists in SAP Identity Management, it returns the `PersonExternalID` as the MSKEYVALUE to be processed.
- If a user with `MSKEYVALUE = UserAssignment.UserID` exists in SAP Identity Management, it returns the `UserAssignment.UserID` as the MSKEYVALUE to be processed.
This updates existing business partner in SAP Identity Management and adds the account for SAP S/4HANA. There is therefore only one identity in SAP Identity Management representing the SAP S/4HANA business user.
- If none of the above-mentioned checks succeed, return the provided `PersonExternalID`.
This creates a new user in SAP Identity Management with `MSKEYVALUE = PersonExternalID`.

Two Identities One Business User

To centrally manage SAP S/4HANA business users in SAP Identity Management, you should use the SAP S/4HANA connector along with the ABAP or the Business Suite connector. You should always run the ABAP or the Business Suite initial load job before the S/4HANA initial load job.

Business partners are linked to SU01 users through the `UserAssignment.UserID` attribute. This attribute value is the `LogonUID` of the SU01 user. As SAP S/4HANA allows the `UserAssignment.UserID` of a given business partner to be different to its `PersonExternalID`, two person entries might be created in SAP Identity Management (the business partner and its linked SU01 user) for one and the same business user in the SAP S/4HANA system. See the example below.

❖ Example

In your SAP S/4HANA system, you have a business partner with `UserAssignment.UserID = John Smith` and `PersonExternalID = JSmith`. This business partner is linked to a SU01 user with `LogonUID = John Smith`.

1. Run the SAP S/4HANA initial load job. A new user with `MSKEYVALUE = JSmith` will be created in SAP Identity Management.
2. Run the ABAP or the Business Suite initial load job. Another user with `MSKEYVALUE = John Smith` will be created in SAP Identity Management.

As a result, two separate users are created in SAP Identity Management (the business partner `JSmith` and its linked SU01 user `John Smith`) for one and the same business user in the SAP S/4HANA system.

The recommended way to handle this is to run the ABAP or Business Suite initial load job first. A new user with `MSKEYVALUE = John Smith` is then created in SAP Identity Management. When followed by the S/4HANA initial load job, a business partner with `UserAssignment.UserID = John Smith` will be mapped to the previously created SU01 user. As a result, you will have one user (with ABAP and S/4HANA account privileges assigned) in SAP Identity Management for one and the same business user in the SAP S/4HANA system.

⚠ Caution

SAP Identity Management does not support cases where one business user is represented by two person entries. To prevent this, you must always run the ABAP or the Business Suite initial load job before the S/4HANA initial load. When loading business users from SAP S/4HANA, the goal is to have only one person entry in Identity Management with two accounts - one for the SAP S/4HANA system, and one for the ABAP system related to the given S/4HANA.

The SAP S/4HANA system automatically synchronizes changes to a business partner with its linked SU01 user. In SAP Identity Management however, there is no automatic synchronization between the business partner attributes and the linked SU01 user. So if you have "two identities one business user" and you change the first name of the given business partner, the first name will be changed in SAP S/4HANA and will be automatically synchronized with the ABAP back end. The SU01 user in SAP Identity Management will keep its old first name until the next ABAP or the Business Suite initial load job is run. Alternatively, you can implement your own synchronization logic in the Identity Management provisioning framework to keep such values relevant and up to date.

Unlike S/4HANA, the ABAP or the Business Suite initial load job has no logic to check if a SU01 user is linked to an SAP S/4HANA business partner. If a link exists, meaning that you ran the S/4HANA initial load job before the ABAP or the Business Suite initial load, the latter will not update the linked business partner in the Identity Management, but will create another identity instead (see the example above).

4.2.11.3 Creating an S/4HANA Business User

In SAP Identity Management, you can create an SAP S/4HANA business user with a custom job or by using the [Create Identity](#) form in the SAP Identity Management User Interface (UI).

Context

The following section describes creating a business user in the UI when the MSKEYVALUE is equal or different to the MX_S4HANA_USERASSIGNMENT_USERID attribute, and the expected results.

In addition to the regular attributes, note that the [Create Identity](#) form contains two SAP S/4HANA specific attributes on the [Settings](#) tab:

- MX_S4HANA_BUSINESS_PARTNER_ROLE_CODE ([Business Partner Role Code](#) field)
This mandatory attribute holds the business partner role code. For example: **BUP003** role code represents employee, **BBP010** role code represents freelancer and **BBP005** role code represents service provider.
- MX_S4HANA_USERASSIGNMENT_USERID ([User Assignment - User ID](#) field)
This optional attribute holds the link of the business partner to the SU01 user.

For more information, see: [S/4HANA Attributes](#).

📘 Note

When creating an SAP S/4HANA business user, the following three attributes are mandatory: MX_S4HANA_PERSON_EXTERNAL_ID, MX_LASTNAME and

MX_S4HANA_BUSINESS_PARTNER_ROLE_CODE. The DISPLAYNAME and MX_FIRSTNAME are mandatory only if you create the business user in SAP Identity Management UI.

Create User with MX_S4HANA_USERASSIGNMENT_USERID Equal to MSKEYVALUE

Context

You can create an SAP S/4HANA business user with MX_S4HANA_USERASSIGNMENT_USERID equal to the MSKEYVALUE. The SU01 user is then linked to the business partner and will have the same user ID (LogonUID) in the ABAP system.

Depending on whether or not a SU01 user exists, proceed as follows:

SU01 User	Procedure
SU01 user does not exist	1. Create a user in SAP Identity Management UI with User Assignment - User ID equal to the MSKEYVALUE and fill in all mandatory attributes. 2. Assign the ABAP and SAP S/4HANA account privileges to the user and save your changes.
SU01 user exists in the ABAP system but not in SAP Identity Management	1. Create a user in SAP Identity Management UI with User Assignment - User ID and MSKEYVALUE equal to the LogonUID of the SU01 user in the ABAP system. 2. Assign the SAP S/4HANA account privilege to the user and save your changes. 3. Run the ABAP initial load job. This will update the newly created user with ABAP specific attributes. Alternatively, you can run the ABAP initial load job first in order to create the SU01 user in the identity store. Then, using the Change Identity form, provide the MSKEYVALUE as the User Assignment - User ID and assign the SAP S/4HANA account privilege to the user.
SU01 user exists in the ABAP system and in SAP Identity Management	1. Search for the SU01 user, select the Change Identity form and provide the MSKEYVALUE as the User Assignment - User ID. On the Settings tab, provide the mandatory Business Partner Role Code value. 2. Assign the SAP S/4HANA account privilege to the user.

Result: You will have one user in SAP Identity Management. Provisioning changes to the new user can be done through the ABAP/Business Suite connector or the SAP S/4HANA connector.

Create User with MX_S4HANA_USERASSIGNMENT_USERID Different to MSKEYVALUE

Context

It is possible to create an SAP S/4HANA business user with MX_S4HANA_USERASSIGNMENT_USERID different to the MSKEYVALUE. The SU01 user is then linked to the business partner and will have a different user ID (LogonUID) in the ABAP system.

Depending on whether or not a SU01 user exists, proceed as follows:

SU01 User	Procedure
SU01 user does not exist	1. Create a user in SAP Identity Management UI with <i>User Assignment - User ID</i> different to the MSKEYVALUE. For example: <i>User Assignment - User ID</i> = John Smith and MSKEYVALUE = JSmith 2. Assign the SAP S/4HANA account privilege to the user and save your changes. 3. Create another user in SAP Identity Management UI with MSKEYVALUE equal to the value in the <i>User Assignment - User ID</i> in step 1. For example: MSKEYVALUE = John Smith 4. Assign the ABAP account privilege to the user and save your changes.
SU01 user exists in the ABAP system but not in SAP Identity Management	The SU01 user exists in the ABAP system (for example, with LogonUID = John Smith). 1. Create a user in SAP Identity Management UI with <i>User Assignment - User ID</i> different than the MSKEYVALUE. For example: <i>User Assignment - User ID</i> = John Smith and MSKEYVALUE = JSmith 2. Assign the SAP S/4HANA account privilege to the user and save your changes. 3. Run the ABAP initial load job. This will create a new user in SAP Identity Management with MSKEYVALUE = John Smith.

SU01 User	Procedure
SU01 user exists in the ABAP system and in SAP Identity Management	The SU01 user exists in the ABAP system (for example, with LogonUID = John Smith), and in SAP Identity Management (for example, with MSKEYVALUE = John Smith). 1. Create a user in SAP Identity Management UI with User Assignment - User ID different than the MSKEYVALUE. For example: User Assignment - User ID = John Smith and MSKEYVALUE = JSmith 2. Assign the SAP S/4HANA account privilege to the user and save your changes.
Result: You will have two users in SAP Identity Management (the SU01 user and the linked business partner). Provisioning changes to the new users can be done through the ABAP/Business Suite connector and the S/4HANA connector to ensure data consistency between SAP Identity Management, SAP S/4HANA and ABAP.	

4.2.11.4 Modifying and Deleting S/4HANA Business User

When modifying an SAP S/4HANA business user in SAP Identity Management and provisioning the changes, you need to ensure data consistency between SAP S/4HANA, the ABAP system related to the SAP S/4HANA system and SAP Identity Management.

Modifying and deleting SAP S/4HANA business user depends on whether the user assignment between the SAP S/4HANA business partner and the SU01 user is active or non-active.

User Assignment is Active

If the user assignment is active, the type of the SU01 user changes from [User with Classic Address](#) to [User with Business Partner Assignment](#). You can no longer change the MX_S4HANA_USERASSIGNMENT_USERID attribute of the SAP S/4HANA business partner. For more information, see: [S/4HANA Attributes](#).

📌 Note

If the user assignment is active, you can modify certain attributes (other than those starting with MX_S4HANA) using either the ABAP/Business Suite connector, or the SAP S/4HANA connector. For example, if you receive the following warnings:

- **S/4HANA ModifyUserCommand: Cannot modify some of the provided attributes, due to policy restrictions.**, this means that the attributes cannot be modified using the SAP S/4HANA connector. The Modify process will fail, and the changes will not be provisioned to SAP S/4HANA. Those attributes can only be modified using the ABAP/Business Suite connector.
- **You cannot maintain address attribute 'Department' in user administration.**, this means that you are trying to modify an attribute, in this case Department, using the ABAP/Business Suite connector, but it can only be modified using the SAP S/4HANA connector.

If the user assignment is active, you cannot delete an SAP S/4HANA user directly in SAP Identity Management. The only way to do this is to first delete its linked SU01 user in the ABAP backend and then delete the S/4HANA user in SAP Identity Management.

User Assignment is Non-Active

If the user assignment is non-active, you can change the MX_S4HANA_USERASSIGNMENT_USERID attribute of a business partner and link it to a different SU01 user.

However, if you try to change the MX_S4HANA_USERASSIGNMENT_USERID attribute of a business partner with one of another business partner, you will get an error message from the S/4HANA SOAP API informing you that a business partner with this user assignment is already reserved. You will also get a warning from SAP Identity Management informing you that attribute MX_S4HANA_USERASSIGNMENT_USERID will be removed from the given user. This is necessary because attribute MX_S4HANA_USERASSIGNMENT_USERID is essential for data consistency between SAP S/4HANA, the ABAP system related to the SAP S/4HANA system, and SAP Identity Management.

In this case you should either run the SAP S/4HANA initial load job or manually reset the value of the attribute from table `mx_i_old_values` because the old value will remain unmodified in SAP S/4HANA and at the same time will be missing in SAP Identity Management.

If the user assignment is non-active, you can delete an SAP S/4HANA user directly in SAP Identity Management.

4.2.12 Password Provisioning

By using password provisioning, you will ensure that the user has the same password in all applications.

The security policy of your organization may not allow this, or only allow this to a limited number of applications.

📘 Note

The password policy of the different applications must be considered when doing password provision.

📘 Note

Not all applications allow setting the productive password.

Related Information

[Provisioning Productive Instead of Initial Passwords \[page 661\]](#)

4.2.12.1 Provisioning Productive Instead of Initial Passwords

By default, the connectors contained in the SAP provisioning framework provision initial passwords, as far as the target system distinguishes between initial and productive passwords.

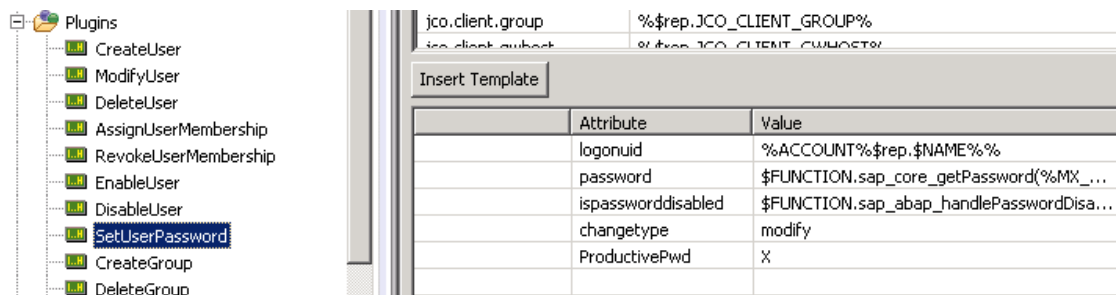
This applies to AS ABAP (including SAP Business Suite and other variations), AS Java and SAP HANA. That means, users typically need to change their passwords when they log on to the target system for the first time after provisioning.

If users set their passwords in the Identity Center, provisioning those passwords as initial passwords does not make sense because there is no use in changing this password in the target system again immediately. For this scenario, most target systems allow setting productive instead of initial passwords when certain configuration prerequisites, such as usage of a secure communication channel, are fulfilled. You can configure the connectors to these target systems to provision productive instead of initial passwords.

AS ABAP

To provision productive passwords to AS ABAP systems, you need to do the following:

1. Apply the SAP Note [1287410](#)
2. [Configure the ABAP Connector to Use SNC \[page 1313\]](#)
3. Set the value of the **ProductivePwd** attribute to **X** on the *Destination* tab of the following passes according to your needs:
 1. [CreateABAPUser](#) - In this case, the user will be created with a productive password.
 2. [ChangePasswordOfABAPUser](#) - In this case, the password of the user will be updated with a productive one.



AS Java

To provision productive passwords to AS Java systems, set the SPML passwordchangerequired attribute in the relevant provisioning jobs to false.

Do so in the AS Java connector package under **Plugins** > **8 Set AS JAVA User password** > **Check backendtype** > **DB** > **SetPasswordJavaUser** > **SetPasswordJavaUser** > **SetPasswordJavaUser** > **ToSPML-USER**.

During the Create task the settings of the security policy of the AS Java are enforced. Therefore, the SPML settings for the password are overruled, and using the Create task you cannot make sure that productive passwords are provisioned.

SAP HANA

Provisioning of productive passwords to SAP HANA is currently not possible.

SUN and Active Directory

Whereas password provisioning for SUN and Active Directory Server repositories only provisions productive passwords. Even if the administrator sets a password this password is provisioned without forcing the user to change his password on the first logon to the back-end system.

4.2.13 Single Sign-On for AS ABAP Systems

If you configured Single Sign-On (SSO) on your AS ABAP source system that you use for the initial load, you assigned a unique SNC name to each user, which is displayed, for example, in the SU01 transaction. The user's SNC name is stored in the table USRACL. You can schedule the **RSUSR300** report to fill this table automatically on a regular basis.

For more information about SNC, see the *SNC User's Guide*.

The initial load jobs of the ABAP Connector and the ABAP BusinessSuite Connector write the user's SNC name to the MX_SNC_Name attribute of the user. If the user is created or modified the SNCName attribute of the connector is filled with the value of the MX_SNC_Name attribute. Users then no longer need to provide manual authentication when logging on to the AS ABAP system.

For more information about how to enable Single Sign-On on the AS ABAP, see *User Authentication and Single Sign-On* in the SAP NetWeaver documentation for your release on the Help Portal.

For an easy to implement SSO solution that integrates well with SAP Identity Management, see *SAP Single Sign-On* on the Help Portal.

Related Information

https://help.sap.com/viewer/p/SAP_SINGLE_SIGN-ON

4.2.14 Preparing the Initial Loads

Context

There are some preparations to make after you have set up the connectors for the systems in your system landscape but before you run the actual initial loads. These include:

Procedure

1. Determining the leading system for attributes.
2. Determine how initial passwords are generated.
3. Run a test initial load.
4. Fix any inconsistencies or other problems.

Related Information

[Determining the Leading System for Attributes \[page 663\]](#)

[Determine How Initial Passwords are Generated \[page 664\]](#)

[Running a Test Initial Load \[page 665\]](#)

[Fix Inconsistencies or Problems \[page 665\]](#)

4.2.14.1 Determining the Leading System for Attributes

Before proceeding, you must determine which system is the leading system for each attribute and role assignment. Then adjust the attributes in the *Destination* tab pages for each write pass in the initial load jobs. Set the period (.) in the first column of the pass definition for attributes in non-leading systems so that these attributes do not overwrite those from the leading one. For role assignments use the {A} option in the pass value if the role assignment is to be added to any existing role assignments. Also adjust the Workflow interface so that these attributes cannot be mistakenly overwritten.

⚠ Caution

This step is very important. If you do not specify the leading system per attribute correctly, attributes could be overwritten from other system, leading to unexpected results.

Example:

For example, the following configuration is for an LDAP directory server that is the leading system for the attributes in the pass. No period is set for these attributes in the WriteABAPUsers pass.

Attribute	Value
MSKEYVALUE	%userid%
MX_MAIL_PRIMARY	%mail%
MX_FIRSTNAME	%givenName%
MX_LASTNAME	%sn%
ACCOUNT%\$rep.\$NAME%	%dn%

In the example below, in the WriteABAPUsers pass for the ABAP initial load, these attributes should not be written to the identity store if the entry already exists. Therefore, the period is set for these attributes. For attributes where the ABAP system is to be the leading system (for example, date format and user type in the example below), no period is set.

In addition, the privilege specified by MXREF_MX_PRIVILEGE is only added if the entry in the identity store does not already exist. Existing role and privilege assignments are not overwritten.

Attribute	Value
MSKEYVALUE	%logonuid%
# changetype	add
ACCOUNT%\$rep.\$NAME%	%logonuid%
MX_DATEFORMAT	%dateformat%
DISPLAYNAME	%displayname%
MX_MAIL_PRIMARY	%primaryMail%
MX_FIRSTNAME	%firstname%
MX_LOCKED	\$FUNCTION.sap_abap_isLocked(%islocked%)\$\$
MX_PASSWORD_DISABLED	\$FUNCTION.sap_abap_isLocked(%ispassworddisabl
MX_LASTNAME	%lastname%
MX_LANGUAGE	\$FUNCTION.sap_abap_convertABAPLangToSOLar
MX_LANGUAGE_COUNTRY	\$FUNCTION.sap_abap_convertABAPLangToSOCot
MX_NUMBERFORMAT	%numberformat%
MX_USERTYPE	%securitypolicy%
MXREF_MX_PRIVILEGE	{A}<PRIV:SYSTEM:%\$rep.\$NAME%>kPRIV:%\$rep.\$

Note

The screen shots above show examples about how the attributes can be set. They do not coincide with the default configuration.

4.2.14.2 Determine How Initial Passwords are Generated

During the initial load or any other task that creates identities in the identity store, the SAP provisioning framework can generate initial passwords for the users. By default, it uses the password policy as specified in the Identity Center. If nothing is specified in the Identity Center password policy, no initial passwords are generated. If you want to specify your own rules, adapt the global Jscript custom_initializePassword.

Optimization option: If you want to ensure that all new identities get some well-defined default values, for example, a default password, and that well-defined workflows are initiated for all new identities, then create a provisioning task that sets the default values and register this task as an [Add event](#) task for the MX_PERSON entry type. This task can also trigger additional Workflow tasks, for example, a task that sends an e-mail.

4.2.14.3 Running a Test Initial Load

You can run a test initial load by creating a temporary identity store to use for the initial load passes. Set the SAP_MASTER_IDS_ID package constant to the temporary identity store. Run the initial loads and check if everything works as expected. Before proceeding with the productive initial load, run the **Reset Delta** job for each repository and change the package constant back to the productive identity store.

See the following sections for typical inconsistencies or errors that you should resolve before performing the initial load.

4.2.14.4 Fix Inconsistencies or Problems

See the sections that follow for possible sources of inconsistencies or problems that may occur during the initial loads. These include the ones listed under Related Links.

Related Information

[Fixing Inconsistencies with Privileges \(AS ABAP\) \[page 665\]](#)

[Out of Memory: Adjusting the Heap Size \[page 666\]](#)

[Timeout: Too many Identities \[page 666\]](#)

4.2.14.4.1 Fixing Inconsistencies with Privileges (AS ABAP)

Context

There may be inconsistencies with privileges on the AS ABAP that lead to errors in the initial load. This can happen due to the order of processing in the initial load, which is:

1. The initial load first reads the AS ABAP roles and creates them in the identity store (in the ReadABAPRoles pass).
2. It then creates the privileges that apply to this role (in the WriteABAPRolePrivileges pass).

3. It then assigns the privileges to the user (in the WriteABAPUsersRolePrivilegeAssignments pass).

Inconsistencies can occur if, for example, the user is assigned to a role or profile that no longer exists. In this case, the initial load cannot read the role or profile and therefore it cannot create the privileges. An error then occurs during the attempt to assign privileges to a user because either the role or privilege does not exist.

If this happens:

Procedure

1. Check whether the role exists on the AS ABAP (use transaction PFCG).
2. If the role does not exist anymore, run the RHAUTUPD_NEW report on the AS ABAP to get a new consistent state. This report eliminates inconsistencies with deleted roles and leftover role assignments.
3. If AUTO_USERCOMPARE is not active, also run the PFCG_TIME_DEPENDENCY report to generate the appropriate profiles.

4.2.14.4.2 Out of Memory: Adjusting the Heap Size

Context

The heap size is specified in the property file for the dispatcher.

If you receive an out of memory error during the initial load due to processing a large number of identities, proceed as follows:

Procedure

1. Set up a separate dispatcher to use for the initial loads.
2. Adjust the heap size for this dispatcher.

4.2.14.4.3 Timeout: Too many Identities

If you receive a timeout during the initial load due to reading a large number of identities from an AS ABAP system, for example, you can one of the following options:

- Activate the bootstrap option for the initial load jobs.
This allows more time for the jobs before a timeout occurs.

- Use the filter mechanism to split up the initial load into several passes. For more information, see [1398976](#).

4.2.15 Running the Initial Loads

Prerequisites

Prerequisites When Using Central User Administration:

1. Make sure that all data is synchronized in the CUA, for example, company address data. To do this, execute the transaction SCUG in the central system.
2. Remove any unnecessary CUA entries that may exist in CUA tables. To do this, execute the RSDELCUA report. Activate the option *Invalid Content in CUA Tables*.
3. Make sure role assignments are up-to-date by executing the user master record comparison (sometimes referred to as text comparison) function in the CUA master system. Execute it for all child systems and activate the *Delete invalid assignments* option.
4. Clean up profiles that are not assigned to any roles by executing the PFUD transaction in the master system. Select the *Cleanups* option.

Context

→ Tip

If you are running an initial load in an already existing productive environment, deactivate provisioning on the dispatcher so that the data read is not provisioned into the various systems. Reactivate provisioning on the dispatcher once the initial load has been completed.

Procedure

1. Activate the *Bootstrap job* option and choose *Apply*. With this option, the bootstrap timeouts are applied to the job instead of the normal job timeouts, giving more time for longer job runs.
2. Run the initial loads for your systems. Select each job and choose *Run now*.

⚠ Caution

Make sure you run the jobs in the correct order.

4.2.16 Cleaning up the Collected Data

After performing the initial loads, the identity data from all systems is stored in the Identity Center's identity store. It is likely that the quality of this data is quite low. Attributes may be duplicated or missing in some sources, or there may be conflicts between attributes. For example, an identity may be represented in several sources by different user IDs, or different identities may be represented in different sources using the same ID. You therefore need to consolidate and clean up this low-quality data and resolve any conflicts before continuing with the provisioning process.

Note

When resolving the data for the use cases described in this document, the user ID is the determining attribute for the identity. This means that each unique user ID that is read from the various data sources is identified and used as the criteria for creating and maintaining identities in the system that is provisioned to.

4.3 Assigning the System Account Privilege to Users or Company Addresses

To specify in which connected systems a user has an account or a company address exists, there is the system account privilege. This system account privilege has the syntax **PRIV:<Repository>:ONLY**.

During an initial load for a repository, this system account privilege is automatically assigned to the identities and company addresses read from the corresponding system. However, when creating identities, or if you want to provision identities to other connected systems, assign the corresponding system account privileges to the identity. To provision a company address to other connected systems, assign the corresponding system account privileges to the company address.

4.4 Using an Update Job for Reading New Privileges and New Company Addresses

Set up this job to run occasionally to update data from the connected systems. This job checks for new privileges and new company addresses in the connected system. These privileges and addresses are then read into the Identity Center.

This update job does not delete privileges that have been deleted in the back-end systems.

A prerequisite for using an update job is that the initial load job has already been executed for the repository before the update job is started.

4.4.1 Creating an Update Job

Prerequisites

- The package is checked out.

Context

To create an update job, proceed as follows:

Procedure

1. In the Identity Management Developer Studio tree view, select the connector package you want to create an update job for. For example: the ABAP connector package [com.sap.idm.connector.abap](#).
2. In the [Jobs](#) folder, copy the Initial Load job and paste it in the same [Jobs](#) folder.
3. Rename the job following the syntax: [<system> - Update](#). For example: [ABAP - Update](#).
4. Adjust the job. The following describes an ABAP or Dual Stack update job. You can modify the initial load job of other systems in a similar manner.
5. In the jobs view, leave only the following passes active:
 - ReadABAPRoles
 - ReadABAPProfiles
 - ReadABAPCompanyAddress
 - ReadJavaRoles
 - WriteABAPRolePrivileges: only if corresponding Read pass is active
 - WriteABAPProfilePrivileges: only if corresponding Read pass is active
 - WriteABAPCompanyAddress: only if ReadJavaRoles pass is active
 - WriteJavaRolePrivileges: only if corresponding ReadJavaRoles pass is active

For each of the other passes, you need to choose [Disable Pass](#) from the context menu.

Note

Make sure that you set the triggers for the privileges in the write tasks. If you upload a privilege and assign it to a user before the AddTrigger executes, the privilege might not be assigned properly. Then, for example, the Identity Center displays the assignment of the privilege to the user, but in the back-end system, no such assignment exists, because the provisioning was not triggered.

Therefore, set the triggers in the write tasks, for example, as follows:

Triggers for Write Tasks: Sample

Write Task	Trigger
MX_ADD_MEMBER_TASK	{D}
MX_DEL_MEMBER_TASK	{D}
MX_PROVISIONTASK	{D}
MX_DEPROVISIONTASK	{D}
MX_VALIDATE_ADD_TASK	{D}
MX_VALIDATE_DEL_TASK	{D}
MX_MODIFYTASK	-1

6. Choose [Save](#).

4.4.2 Scheduling an Update Job

Context

Depending on how often you add new privileges in the back-end systems, schedule the update jobs to run frequently, for example, daily.

To schedule the job:

Procedure

1. Select the update job you created for each system that should be updated.
2. On the [General](#) tab, select the schedule rule that applies for this job. For example: [Midnight](#).
3. Choose [Save](#).

4.5 Assigning Function Set Privileges to Users for SAP Business Suite Applications

Context

When working with the enhanced SAP Business Suite use case, function set privileges are created for each of the SAP Business Suite applications that are activated for provisioning with SAP Identity Management. These privileges have the following syntax:

```
PRIV:FUNCTION_SET:<Repository>:<BAdI_Filter_Value>
```

Procedure

Assign this privilege to business roles or identities that use the corresponding SAP Business Suite application.

Note

Active SAP Business Suite applications are specified in the `IDM_BADI_FILTER` BAdI filter table.

4.6 Setting up an SAP Java Connector (SAP JCo) and Related Traces

To analyze errors related to an issue with a connection from the Identity Center to an AS ABAP, you must first activate traces to get more information.

Context

These traces log information for all ABAP-related passes of the relevant dispatcher. In connection with JCo the following traces are available: JCo trace, JRFC trace, and CPIC trace. In this order they contain more and more technical details.

Note

We recommend that you start the analysis using the JCo trace. Only if this does not collect enough technical details, use the JRFC or even the CPIC trace.

To set up the SAP JCo related tracing, you can either configure tracing for all ABAP-related passes of the relevant dispatcher using Java system properties or you can just trace specific passes. For more information on tracing specific passes, see the *Related Links* section.

→ Tip

For performance reasons and to avoid unnecessary amounts of data in the traces, make sure that you only trace as much as necessary. Rather use pass-specific traces than traces for the complete dispatcher.

For more information about analysis of JCo traces in general, see *JCo Exceptions* in the SAP Library for your AS Java release.

Procedure

1. In the Identity Management Developer Studio tree view, select the *Dispatchers* folder (under *Management* in your Identity Management database).
2. Select a dispatcher and choose the *Policy* tab.
3. In the *Java options* field, enter the global trace parameters for JCo. Make sure to place **-D** in front of every single parameter and separate multiple options with a space character.

The table below shows the available traces going from the most global one to the one containing most technical details.

Global Trace Parameters

Parameter	Possible Values	Description
jco.trace_path	Enter an existing path.	Sets the location of the path for JCo trace files. To output the JCo trace into a file, set the path value, to an existing directory. If the directory does not exist, no log files are written. Note If there are spaces in the path name, make sure that you use quotation marks ("), for example: <pre>-Djco.trace_path="C:\usr\sap\IdM\Identity Center" -Djco.trace_level=8 -Djrfe.trace=1</pre> For each event a trace is written into a log file with the following syntax: JCO<date>_<timestamp>.trc Available for JCo3 and JCo2.
jco.trace_level	Valid values: levels 0 to 10. The commonly used ones are described in the Trace Levels table below. Default value: 0	Turns on the JCo trace. Available for JCo3 and JCo2.

Parameter	Possible Values	Description
jrfc.trace	0: trace is off 1: RFC trace is on Default value: 0	If set to 1, RFC trace is turned on for all connections within JCo. The trace is written to the location as indicated in jco.trace_path. For each event a trace is written into a log file with the following syntax: <pre>jco_rfc<identifier>_<number>.trc</pre>

Note

For performance reasons, only use this parameter in case of emergency to activate an RFC trace. Rather activate the RFC trace for specific passes.

By default, the RFC trace is disabled. The destination name is the repository name.

Available for JCo3 only.

cpic.trace	4 Valid Values: 0 to 3 0: trace is off 3: most verbose trace Default value: 0	The trace level for the CPIC layer. The CPIC trace is always written into the Identity Center installation directory, for example, \usr\sap\IdM\Identity Center. Depending on where the trace is configured, the file name is: - CPIC<date>_<Timestamp>.trc, when configured globally in the dispatcher. - CPICTRC<identifier>, when configured in the pass.

Available for JCo3 only.

Trace Levels

Value	Description
0	No trace.
1	Traces errors.
2	Traces errors and warnings.

Value	Description
3	Traces info messages, errors, and warnings.
4	Traces execution path, info messages, errors, and warnings.
5	Verbose trace of execution path, info messages, errors, and warnings.
6	Verbose trace of execution path, limited data dumps, info messages, errors, and warnings.
7	Full trace of execution path, data dumps with metadata, verbose info messages, errors, and warnings.
8	Full trace of execution path, full data dumps with metadata, verbose info messages, errors, and warnings.

4. To save your changes, choose [Apply](#).
5. Regenerate the dispatcher scripts.
6. For the changes to take effect, restart the relevant dispatchers.
7. For performance reasons, deactivate the traces as soon as possible.

Related Information

[JCo Exceptions](#). This topic describes the exceptions for an SAP NetWeaver AS for Java. The properties of the exceptions basically also apply for the Identity Management use case. The file locations differ.

[Restricting the CPIC or JRFC Trace to a Specific Pass \[page 676\]](#)

4.6.1 Restricting the CPIC or JRFC Trace to a Specific Pass

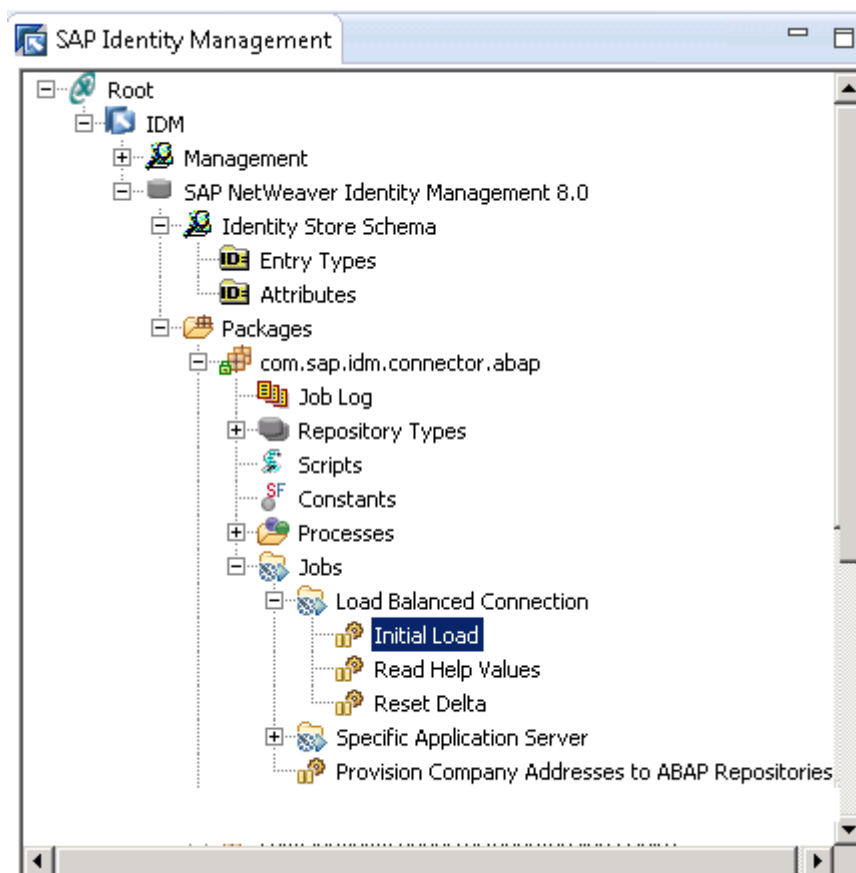
Identity Management system administrators can configure CPIC and JRFC traces (only trace level) for specific ABAP-related passes with an additional parameter in the From/ToCustom passes with pass types FromSAP, FromSAPIdentity, ToSAP and ToSAPIdentity. Then only those connections that cause problems are traced.

Context

To add a parameter and its value, proceed as follows:

Procedure

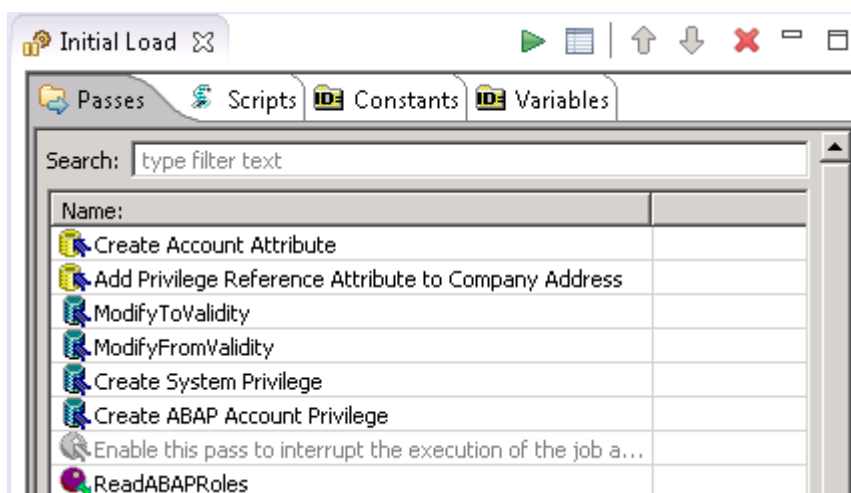
1. In the tree view in the Identity Management Developer Studio, select your identity store and then *Packages* folder.
2. Select a package that contains ABAP-related passes. For example, *com.sap.idm.connector.abap* or *com.sap.idm.connector.abap.businesssuite*.



3. In the *Jobs* folder, select a job and choose *Open* from the context menu.

This opens the job in the job view.

4. In the job view, select the *Passes* tab.
5. Select the pass you want to add a parameter to and its value, and choose *Open* from the context menu.



Note

This is also valid for the ABAP BusinessSuite Connector passes of the listed types in standalone jobs like initial load and read help values (actually: ABAP - Initial Load, ABAP - Provision Company Addresses to ABAP Repositories, ABAP Read Help Values, BusinessSuite AS ABAP - Initial Load) as well as for custom provisioning jobs and standalone jobs using these pass types.

6. Choose the *Destination* tab for provisioning job passes or the *Source* tab for standalone job passes.
7. To add a parameter and its value, select an existing line in the parameter-value table and choose *Insert Row* from the context menu.

Note

The package must be checked out and ready for editing before you can change the properties.

8. Enter the parameters using the following data:

Trace Parameters

Parameter	Value	Comment
jco.client.trace	1	This is the pass-specific variant of the jrhc.trace parameter.
jco.client.cpic_trace	3	This is the pass-specific variant of the cpic.trace parameter.

General	Source	Destination	Delta	Documentation

Source - Custom

Pass Type: FromSAP

Parameter	Value:
RETURN_EMPTY_VALUES	false
ENTRYTYPE	ROLE
FILTER	*
jco.client.trace	1
jco.client.cpic_trace	3
jco.client.ashost	%%\$rep.JCO_CLIENT_ASHOST%
jco.client.sysnr	%%\$rep.JCO_CLIENT_SYSNR%
jco.client.user	%%\$rep.JCO_CLIENT_USER%
jco.client.client	%%\$rep.JCO_CLIENT_CLIENT%
jco.client.passwd	%%\$rep.JCO_CLIENT_PASSWD%
jco.client.lang	%%\$rep.JCO_CLIENT_LANG%
jco.client.group	%%\$rep.JCO_CLIENT_GROUP%
jco.client.gwhost	%%\$rep.JCO_CLIENT_GWHOST%
jco.client.gwserv	%%\$rep.JCO_CLIENT_GWSERV%
jco.client.mshost	%%\$rep.JCO_CLIENT_MSHOST%
jco.client.r3name	%%\$rep.JCO_CLIENT_R3NAME%
jco.client.snc_lib	%%\$rep.JCO_CLIENT_SNC_LIB%
jco.client.snc_mode	%%\$rep.JCO_CLIENT_SNC_MODE%
jco.client.snc_myname	%%\$rep.JCO_CLIENT_SNC_MYNAME%
jco.client.snc_partnern...	%%\$rep.JCO_CLIENT_SNC_PARTNERNAME%
jco.client.snc_qop	%%\$rep.JCO_CLIENT_SNC_QOP%
CUA_MASTER	%%\$rep.CUA_MASTER%

- To save your changes, select the package in the tree view and choose [Check in](#) from the context menu.

5 Connecting Non-SAP Systems

This section describes how to connect non-SAP systems to SAP Identity Management and set up your landscape accordingly.

Related Information

[Setting Up a Microsoft Active Directory and Microsoft Exchange \[page 679\]](#)

[Setting Up a Lotus Notes System \[page 682\]](#)

5.1 Setting Up a Microsoft Active Directory and Microsoft Exchange

The Active Directory (AD) connector makes it possible for the SAP Identity Management to provision users and groups to and from Microsoft Active Directory. You can also use it to create and mail-enable users in Microsoft Exchange 2007 or Microsoft Exchange 2010.

Prerequisites

The following software is required:

- SAP Identity Management 8.0 SP00 (or higher) is correctly installed and licensed.
- Microsoft Active Directory versions with Microsoft Windows Server 2000/2003/2008/2012/2012 R2/2016/2019/2022
- Microsoft Exchange 2007/2010
- Platform: MS Windows Server 2000/2003/2008/2012/2012 R2/2016/2019/2022
- You are familiar with the restrictions that apply to all connectors and to LDAP Directory connectors only, described in:
 - [Restrictions that Apply to All Connectors \[page 1332\]](#)
 - [Restrictions for LDAP Directory Connectors \[page 1335\]](#)

Overview

The AD connector is used for communication between SAP Identity Management and Microsoft Active Directory. SAP Identity Management can provision users and groups to and from Microsoft Active Directory

using `To LDAP directory` and `From LDAP directory` passes. The connector is delivered as a separate package in the Provisioning framework for SAP Identity Management 8.0. The name of the package is [com.sap.idm.connector.ad](#).

The connector supports the following use cases:

- Loading users and groups with attributes
- Create, modify, enable, disable and delete users
- Create and delete groups
- Assign and unassign users to ADGroup

To configure the AD connector, you need to:

1. Import the AD connector package. See [AD Connector Package \[page 445\]](#) and [Importing the Provisioning Framework for SAP Identity Management 8.0 \[page 487\]](#).
2. Create a repository of type *Active Directory* with the correct values for the repository constants. See [Creating Repositories \[page 493\]](#) and [Repository Constants for Active Directory for Provisioning Framework \[page 1256\]](#).
3. Run an Initial Load job. See [Running the Initial Loads \[page 667\]](#).
Once you run the *Initial Load* job, the connector loads the users and groups to SAP Identity Management according to the mapping between Identity Management and Microsoft Active Directory attributes. See [Mapping Between Identity Management and Active Directory Attributes \[page 1304\]](#).

You can use the Provisioning framework for SAP Identity Management 8.0 to create and mail-enable users in Microsoft Exchange 2007 or Microsoft Exchange 2010. See the Related Information section.

Related Information

[Creating and Mail-Enabling Users in Microsoft Exchange \[page 680\]](#)

5.1.1 Creating and Mail-Enabling Users in Microsoft Exchange

You can use the Provisioning framework for SAP Identity Management 8.0 in order to create and mail-enable users in Microsoft Exchange 2007 or Microsoft Exchange 2010.

Prerequisites

- You have imported the AD connector package.
- You have created a repository of type *ActiveDirectory*.

Context

In SAP Identity Management 8.0, the [CreateUser](#) plugin process of the AD connector package contains the [CreateExchange2007User](#) and [CreateExchange2010User](#) action tasks that are enabled by default. Both action tasks contain a [To LDAP directory](#) pass, that is [Mail enable the user](#).

To configure the creating and mail-enabling users in Microsoft Exchange, you need to complete the following steps:

Procedure

1. Create a Microsoft Exchange user. That is, you need to create a mailbox for an Active Directory user. This mailbox will be used as a template mailbox.

Note

The template mailbox is mandatory. It can be disabled and hidden in the Active Directory, but it has to be created.

- For Microsoft Exchange 2007, you need to create a mailbox for an Active Directory user and then use the `TemplateInstance` parameter to point to this mailbox so that it can be used as a template mailbox.
For more information, see *How to Use Templates to Create Recipients* in Microsoft Exchange Server 2007 documentation.
 - As of Microsoft Exchange 2010, the `TemplateInstance` parameter is not supported. You need to create a standard mailbox for an Active Directory user that will take the role of a template mailbox.
2. Set the repository constants for creating Exchange mailboxes. Pay attention to the repository constant `LDAP_TEMPLATE_MAILBOX`.

The `CreateExchange2007User` and `CreateExchange2010User` action tasks depend on a repository definition constant `LDAP_TEMPLATE_MAILBOX` that has to be created in the repository definition defining the Active Directory where the users will be created. The `LDAP_TEMPLATE_MAILBOX` constant should be set to be the DN of the template mailbox, for example `DN: cn=Template Mailbox,cn=Users,DC=example,dc=com`,

3. Configure the Microsoft Exchange attributes.

The attributes added to Active Directory are the attributes that are added by default by the Microsoft Exchange system when enabling a mailbox. The best way to determine which attributes will be needed for your system is to inspect the attributes added to the template mailbox when this is created.

To add new attributes, open the Mail enable the user pass, select the [Destination](#) tab, choose [Insert Row](#) and enter the name of the attribute.

Related Information

[Repository Constants for Active Directory for Provisioning Framework \[page 1256\]](#)

5.2 Setting Up a Lotus Notes System

The Lotus Notes connector makes it possible for SAP Identity Management to provision users and groups to and from Lotus Domino server.

Prerequisites

SAP Identity Management 8.0 SP01 (or higher) is correctly installed and licensed

IBM Lotus Domino server 8.5.3 (or higher)

Notes ID Vault is created and configured

DIOP processes are running on the server

The technical user that is used to establish the connection to the Domino server needs at least Editor access level to the Names.nsf database. This access level allows the technical user to create and edit all documents in the database.

You are familiar with the restrictions that apply to all connectors, described in: [Restrictions that Apply to All Connectors \[page 1332\]](#).

Overview

The Lotus Notes connector is used for communication between SAP Identity Management and IBM Lotus Notes systems. SAP Identity Management can provision users and groups to and from Lotus Domino server using [ToNotes](#) and [FromNotes](#) passes. The connector is delivered as a separate package in the Provisioning framework for SAP Identity Management 8.0. The name of the package is [com.sap.idm.connector.notes](#). The Lotus Notes connector is delivered with its own Lotus Notes schema.

As of SAP Identity Management 8.0 SP01, the Lotus Notes connector is Java-based, compared to SAP NetWeaver Identity Management 7.2 where the connector uses Windows Runtime.

The connector supports the following use cases:

- Loading users and groups with attributes
- Creating, modifying and deleting users and groups

ⓘ Note

Password is required during user creation.

ⓘ Note

User ID files are created in Notes ID Vault and attached to person documents during user creation.

- Changing user password

Note

Only standard user password is supported, not internet password.

- Renaming and re-certifying users
- Assigning and unassigning members to group
- Only Notes ID Vault systems are supported

To configure the Lotus Notes connector, follow the steps below in the exact order:

1. [Import the Lotus Notes Schema \[page 684\]](#).
2. [Import the Lotus Notes Connector Package \[page 686\]](#).
3. [Add NCSO.jar in the Identity Management Dispatcher Classpath \[page 687\]](#). This file contains the Java API for Domino servers.
4. [Adjust the Attribute Values \[page 688\]](#) for MX_NOTES_MAILSYSTEM, MX_SALUTATION, MX_LANGUAGE and MX_NOTES_GROUP_GROUPTYPE attributes.
5. Create a repository of type *Notes* with the correct values for the repository constants. For more information, see [Creating Repositories \[page 493\]](#) and [Repository Constants for Lotus Notes \[page 1263\]](#).
6. [Set Up the Replication Source for the Mail File \[page 689\]](#).
7. Run an Initial Load job. For more information, see [Running a Job \[page 297\]](#).
Once you run the Initial Load job, the connector loads the users and groups to SAP Identity Management according to the mapping between Identity Management and Domino Technical attributes. See [Mapping Between Identity Management and Domino Technical Attributes \[page 1301\]](#).
8. [Modify Notes Certifiers \[page 691\]](#).

Related Information

[Lotus Notes Connector Package \[page 468\]](#)

[Configuring Password Reset \[page 690\]](#)

[Recertifying Notes Users \[page 691\]](#)

[Adding New Attributes \[page 692\]](#)

[Troubleshooting: User Creation Fails in Lotus Domino Server](#)

5.2.1 Importing the Lotus Notes Schema

Context

To import the Lotus Notes schema, proceed as follows:

Procedure

1. In the Identity Management Developer Studio tree view, select the *Identity Store Schema* node of your identity store and choose *Import* from the context menu.
2. Select the `LotusNotes_Schema.idmschema` file that is located in `<Install_folder>\Core\ConfigurationPackages\LotusNotes` and choose *Open*.

The Lotus Notes schema is added to the identity store schema.

Related Information

[Lotus Notes Schema \[page 684\]](#)

5.2.1.1 Lotus Notes Schema

The identity store stores the identity data according to a schema that consists of entry types and attributes. The entry types are objects that describe how the different identity-relevant objects are represented in the Identity Center. In SAP Identity Management 8.0, the Lotus Notes connector is delivered with a schema that consists of the following entry types:

Entry type SAP_NOTES_CERTIFIER

This entry type is used to hold Notes certifier. The following attributes are allowed for this entry type:

Attributes	Mandatory (Yes/No)	Available as of SAP Identity Management version	
		Management version	Description
DISPLAYNAME	Yes	7.2	Display name of the entry.

Attributes	Mandatory (Yes/No)	Available as of SAP Identity Management version	Description
MSKEYVALUE	Yes	7.2	Unique entry identifier.
MX_NOTES_CERTIFIER_FILE	No	7.2 SP5	The certifier file used for creating the user. The certifier file must be on the Domino server.
MX_ENCRYPTED_PASSWORD	No	7.2	The certifier password for the certifier file.
MX_ENTRYTYPE	Yes	7.2	Type of entry.

Entry type SAP_NOTES_SERVER

This entry type is used to hold Notes servers that are part of a cluster. The following attributes are allowed for this entry type:

Attributes	Mandatory (Yes/No)	Available as of SAP Identity Management version	Description
DISPLAYNAME	Yes	7.2	Display name of the entry.
MSKEYVALUE	Yes	7.2	Unique entry identifier.
MX_ENTRYTYPE	Yes	7.2	Type of entry
MX_NOTES_MAILSERVER	No	7.2 SP5	The IP address of the mail server.
MX_NOTES_MAILSYSTEM	No	7.2 SP5	The users mail system, such as Lotus Notes, CcMail, Vim.
MX_NOTES_OWNER	No	7.2 SP5	Owner of the Lotus Notes object.

Entry type SAP_NOTES_POLICY

This entry type is used to hold Notes policy. The following attributes are allowed for this entry type:

Attributes	Mandatory (Yes/No)	Available as of SAP Identity Management version	Description
DISPLAYNAME	Yes	7.2	Display name of the entry.
MSKEYVALUE	Yes	7.2	Unique entry identifier.
MX_ENTRYTYPE	Yes	7.2	Type of entry.
MX_NOTES_FULLNAME	No	7.2 SP5	Entry's full name.

5.2.2 Importing Lotus Notes Connector Package

Prerequisites

- You have imported the engine package and the custom package. For more information about the recommended order to import packages, see *Package Dependencies*.
- You have imported the Lotus Notes schema.

Context

To import the Lotus Notes connector package, proceed as follows:

Procedure

1. In the Identity Management Developer Studio tree view, select the [Packages](#) folder of your identity store and choose [Import](#) from the context menu.
2. Select the `com.sap.idm.connector.notes.idmpck` file that is located in `<Install_folder>\Core\ConfigurationPackages\LotusNotes` and choose [Open](#).
3. In the [Enter Import Reason](#) dialog box, you can do the following:
 - (Optional) Enter a reason for the import.
 - Choose [Import](#).
The package is imported with its package qualified name.

- Choose [Import as New Package](#) and enter a name for the package.
The package is imported under a new name. You can use this option if you want to modify the Lotus Notes connector package and keep your changes when updating the connector package.
4. Choose [OK](#).

Results

The package is imported into the Identity Management Developer Studio. The [Status](#) on the [Dependencies](#) tab of the imported package should be [Linked OK](#).

Related Information

[Importing the Lotus Notes Schema \[page 684\]](#)

[Package Dependencies \[page 404\]](#)

5.2.3 Adding NCSO.jar in Identity Management Dispatcher Classpath

Context

You need to add the `NCSO.jar` file in the Identity Management dispatcher classpath. The `NCSO.jar` file contains the Java API for Domino servers.

Procedure

1. Copy the `NCSO.jar` file to the Identity Center home Java subfolder. For example:
`C:\usr\sap\idm\Identity Center\Java`

This file can be found in the Lotus Domino server installation. For example:
`c:\lotus\domino\data\domino\java\NCSO.jar`
2. Navigate to the Identity Center home `./Service-Scripts` and open `<dispatcher_name>.props` file.
3. Change the property [DSECLASSPATH](#) to add the path to the `NCSO.jar`.
For example: `DSECLASSPATH=%DSE_HOME%/Java/DSE.jar;%DSE_HOME%/Java:%DSE_HOME%/Java/sapjco.jar;%DSE_HOME%/Java/sapjco3.jar;%DSE_HOME%/Java/NCSO.jar`

5.2.4 Adjusting Attribute Values

You need to add legal values for MX_NOTES_MAILSYSTEM, MX_SALUTATION, MX_LANGUAGE and MX_NOTES_GROUP_GROUPTYPE attributes. In addition, use the MX_NOTES_MAILSYSTEM attribute values in the sap_notes_convertMailSystemLoad script accordingly.

Adding Legal Attribute Values

Context

To add legal values for MX_NOTES_MAILSYSTEM, MX_SALUTATION, MX_LANGUAGE and MX_NOTES_GROUP_GROUPTYPE attributes, proceed as follows:

Procedure

1. In the Identity Management Developer Studio tree view, select the *Identity Store Schema* of your identity store and then select *Attributes*.
2. Select an attribute and choose *Open* from the context menu.
3. Select the *Attribute Values* tab and enter legal values in the *Value* field.

Note

The attribute values should be the same as on the Lotus Notes system. For example:

MX_NOTES_MAILSYSTEM attribute holds the information about the user's mail system. Legal values for this attribute are: Lotus Notes, CcMail, Vim, and so on.

MX_NOTES_GROUP_GROUPTYPE attribute holds the information about the Lotus Notes group types. Legal values for this attribute are: 0 (Access Control List only), 1 (Deny List only), 2 (Mail only), 3 (Multi-purpose), 4 (Server only) and so on.

4. Choose *Save*.

Adding Values for sap_notes_convertMailSystemLoad Script

Context

To add legal values for sap_notes_convertMailSystemLoad script, proceed as follows:

Procedure

1. In the Identity Management Developer Studio tree view, select the Lotus Notes connector package [com.sap.idm.connector.notes](#) and then select [Scripts](#).
2. Select the sap_notes_convertMailSystemLoad script and choose [Edit](#) from the context menu.
3. Enter the same values as for MX_NOTES_MAILSYSTEM attribute.

Bear in mind the numeric values, for example: 0 = NOTES; 1 = CCMAIL

4. Choose [Save](#).

5.2.5 Setting Up the Replication Source for the Mail File

Context

You can replicate a user's mail file by using the Domino Registration server as a source (the location from which the mail file will be copied). However, if you have separate Registration and Mail servers, make sure the Mail server is specified as the replication source for the mail file of the user.

Mail file replication may fail if you have separate Registration and Mail servers and the above mentioned condition is not met.

To set up the source for the mail file, proceed as follows:

Procedure

1. In the Identity Management Developer Studio, expand the [com.sap.idm.connector.notes](#) package and select [Processes > Plugins > CreateUser](#).
2. Select the [Create Notes User](#) action task and open the pass properties.
3. On the [Destination](#) tab, search for the [ReplicationSourceServer](#) attribute in the [Definitions](#) table. Set its value to be the Mail server (the location from which the mail file will be copied).
4. Save your configuration by choosing [Ctrl](#) + [S](#).

5.2.6 Configuring Password Reset

Context

Note

Before you start configuring the password reset for Lotus Notes users, have in mind some specifics of the Lotus Notes Domino server.

The Java API of the Domino server does not allow the Password Reset method to be called remotely. That is why this method should be called from an agent running on the Domino server. The following process takes place:

1. The Lotus Notes connector sends request to the Domino agent by passing three parameters (server, user, password) to the agent.
2. The Domino agent gets the parameters and calls the Password Reset method in the Java API of the Domino server.

To configure the password reset, proceed as follows:

Procedure

1. Create a Domino agent. For more information, see *Implementing Password Reset Agent for IdM Lotus Notes Connector* on the SCN.

If you already have an agent running on the Domino server, skip this step.

2. In SAP Identity Management Administration User Interface, open a repository of type **Notes** and enter values for the repository constants `PASSWORD_RESET_AGENT_DATABASE` and `PASSWORD_RESET_AGENT_NAME`.
3. Use the [Modify User](#) process to change the password.

Related Information

[Implementing Password Reset Agent for IdM Lotus Notes Connector](#) 

5.2.7 Modifying Notes Certifiers

You can modify Notes certifiers by using a form (UI task) that is delivered with the Lotus Notes connector.

Context

Once you have run the Initial Load job, certifiers are loaded into the identity store. For every certifier you need to enter the certifier ID file path on the server and the password for the certifier ID file.

To modify Notes certifier, proceed as follows:

Procedure

1. Log on to SAP Identity Management User Interface at <http://<host>:<port>/idm>.
2. On the [Manage](#) tab, select [SAP_NOTES_CERTIFIER](#) from the [Show](#) list.
3. In the [Find](#) field, enter the name of the certifier you want to modify and choose [Go](#).
4. Select the certifier and choose [Choose Task](#).
5. In the [Choose Task](#) dialog box, choose [Lotus Notes Forms](#) > [Modify Certifier](#) > [Choose Task](#).
6. In the [Certifier](#) field, enter the certifier id file path on the server.
7. In the [Encrypted Password](#) field, enter the password for the certifier id file.
8. Choose [Save](#).

5.2.8 Recertifying Notes Users

You can recertify Notes users by using a form (UI task) that is delivered with the Lotus Notes connector.

Context

To recertify a Notes user, proceed as follows:

Procedure

1. Log on to SAP Identity Management User Interface at <http://<host>:<port>/idm>.
2. On the [Manage](#) tab, select [Person](#) from the [Show](#) list.
3. In the [Find](#) field, enter the name of the Notes user that you want to recertify and choose [Go](#).

4. Select the user and choose *Choose Task*.
5. In the *Choose Task* dialog box, choose ► *Lotus Notes Forms* ► *Re-certify Identity* ► *Choose Task* ►.
6. In the *Expiration Date* field, specify a valid date as in the following format: **11/04/2015**.
7. Choose *Save*.

5.2.9 Adding New Attributes

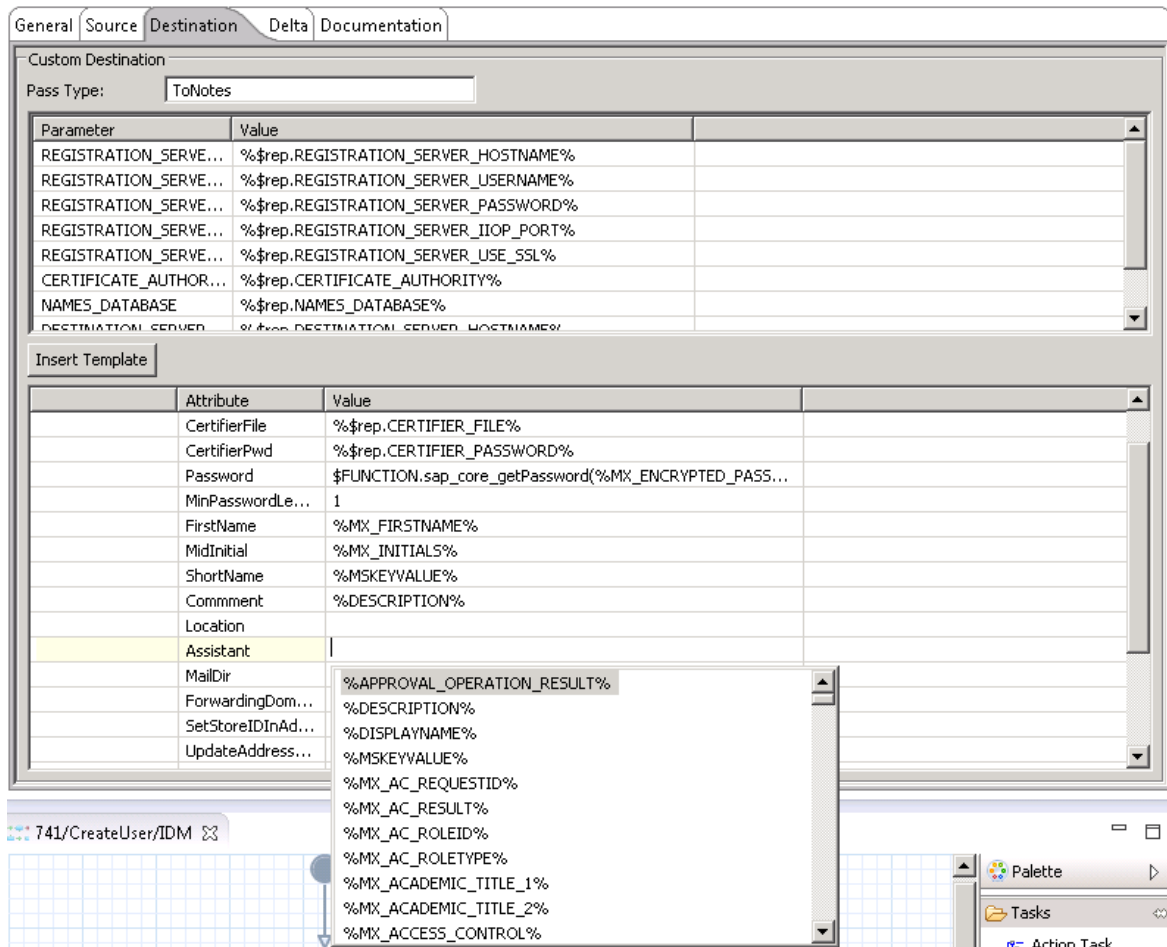
You can extend the number of attributes for users and groups that are provisioned to a Lotus Notes system. This is applicable for creating and modifying users, as well as for creating groups.

Context

To add new attributes to users, proceed as follows:

Procedure

1. In the tree view, select *Create User* process and choose *Open* from the context menu.
2. Select *Create Notes User* pass and choose *Open Pass Properties* from the context menu.
3. Select the *Destination* tab.
4. In the definitions table, choose *Insert Row* from the context menu.
5. Enter the name of the attribute (for example, **Assistant**) and insert the attribute value by selecting an attribute from the list in the context menu.



6. Choose **Save**.

5.2.10 Migrating to Java-Based Lotus Notes Connector

Migrating to the Java-based Lotus Notes connector in SAP Identity Management 8.0 is relevant only for an upgrade scenario from SAP Identity Management 7.2 to 8.0. You need to run a **Migrate Data from 7.2** job and adjust a number of attribute mappings in the **Initial Load** job.

Prerequisites

- You have upgraded from SAP Identity Management 7.2 to 8.0.
- You have imported the Lotus Notes schema.
- You have imported the Lotus Notes connector package.

Context

When you migrate to the Java-based Lotus Notes connector, you need to run the **Migrate Data from 7.2** job to migrate all certifier, policy and server entries from your 7.2 system to your 8.0 system. The job is delivered with three **To Identity store** passes that write the certifier, policy and server entries in the identity store in the Identity Management database.

You also have to adjust a number of attribute mappings in the group-related passes in the **Initial Load** job.

Procedure

1. In the Identity Management tree view, select the Lotus Notes connector package and run the **Migrate Data from 7.2** job.
2. In the **Job** view, open the **Initial Load** job.
3. For the group-related passes, select the **Destination** tab and remove the **GROUP** prefix from the attribute mappings.

In the **WriteGroups** pass for example, mapping **MSKEYVALUE = GROUP:%ListName%** should become **MSKEYVALUE = %ListName%**.

In the table below, see the valid attribute mappings for the group-related passes:

Pass Name	Attribute Name	Value
WriteGroups	MSKEYVALUE	%ListName%
Add owners to group	MSKEYVALUE	%refid%
Add members to groups	MSKEYVALUE	%refid%
Create Notes Group Privileges	MXMEMBER_MX_GROUP	{A}<%ListName%>
Write group changenumber	MSKEYVALUE	%ListName%
	SAP_CHANGENUMBER	%ListName%

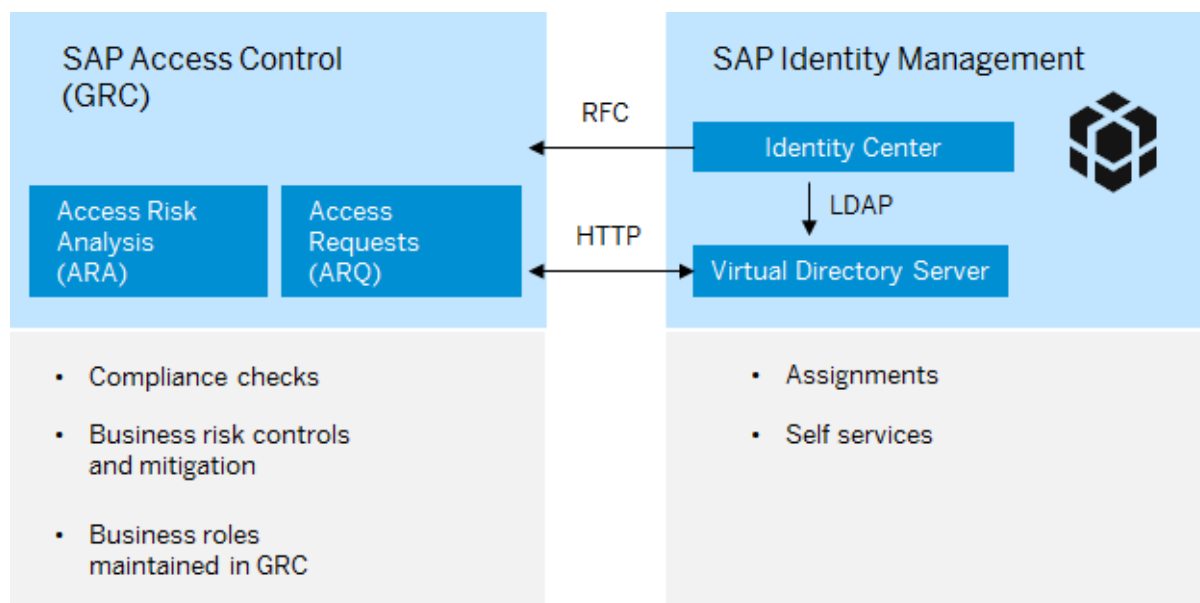
4. Run the **Initial Load** job.

6 Integration with SAP Access Control 10.0 or higher

This section gives an overview of the integration between SAP Identity Management 8.0 and SAP Access Control (for SAP Governance, Risk and Compliance (GRC) Solutions) 10.0 or higher using the GRC 10.0 or higher Provisioning Framework Version 2.

The SAP Identity Management GRC integration consists of a set of processes in the Identity Center and a configuration in the Virtual Directory Server that enables the use of SAP Access Control for risk validation before user provisioning. Using this solution, SAP Identity Management can execute provisioning to multiple target systems which are controlled by SAP Access Control to ensure compliance according to the rules implemented here.

When business requirements imply compliancy and Segregation of Duties checks, SAP Identity Management performs risk validation on SAP Access Control before assigning permissions, in order to achieve the compliant provisioning.



Depending on the communication between SAP Access Control and SAP Identity Management, the components of SAP Identity Management are used in the following way:

RFC Communication

The RFC communication between SAP Access Control and SAP Identity Management is used when implementing Business role synchronization.

- Identity Center:
Runs Initial Load Business Roles job by making an RFC call to GRC.

Web Service Communication

The Web Service communication between SAP Access Control and SAP Identity Management is used when implementing scenarios for technical roles (privileges).

- Virtual Directory Server:
Accepts requests from Identity Center.
Deals with all connection to/from SAP Access Control through the web service API exposed by SAP Access Control.
- Optionally, another Virtual Directory Server deployed as an Identity Service on an AS Java server.
Deals with event-based result handling.
- Identity Center:
Contains the workflow tasks and the necessary jobs that drive the risk validation using SAP Access Control, before provisioning based on the Provisioning Framework for SAP Systems.
Communicates with the Virtual Directory Server (VDS) using the LDAP protocol.

Note

Using both integration scenarios - SAP Identity Management with Central User Administration (CUA) system and SAP Identity Management with GRC, is not supported. You can use only one of them in your system landscape.

Related Information

[Integration Scenarios \[page 696\]](#)

[Configuring the Virtual Directory Server \[page 710\]](#)

[Importing the GRC Provisioning Framework \[page 724\]](#)

[Configuring the Solution \[page 728\]](#)

[Running the Initial Load Service Jobs \[page 750\]](#)

6.1 Integration Scenarios

Several integration scenarios exist, depending on three factors:

- Landscape configuration
- Result handling
- Provisioning initiation

The GRC 10.0 or higher Provisioning Framework Version 2 can be configured to deal with any combination of these scenarios.

Related Information

[Landscape Configuration Scenarios \[page 697\]](#)

[Result Handling Scenarios \[page 700\]](#)

[Provisioning Initiation Scenarios \[page 702\]](#)

[Integration Scenario Combinations \[page 705\]](#)

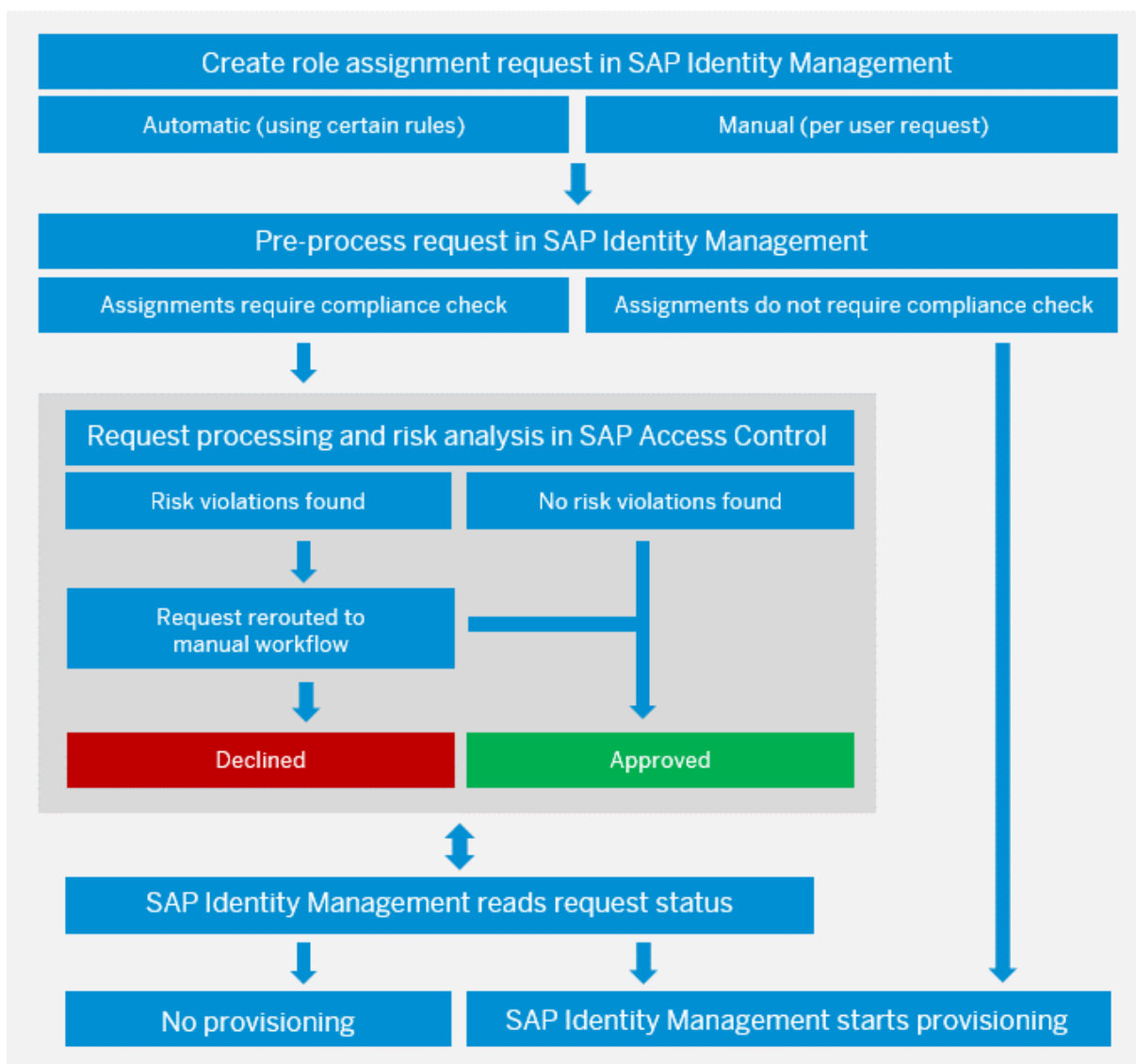
6.1.1 Landscape Configuration Scenarios

There are two landscape configuration scenarios for the integration:

Centralized Provisioning

This is a scenario where SAP Identity Management is the only provisioning system, responsible for provisioning both the assignments that require and do not require compliance checks to the target systems (both SAP and non-SAP).

The SAP Identity Management uses SAP Access Control to execute risk analysis, that is, the privilege requests are sent to SAP Access Control (usually to Access Risk Analysis (ARA)) for risk analysis, while the Identity Management does the provisioning based on the risk analysis result. This means that the Identity Management has at least one ABAP repository and the privileges associated with it (the privileges, some or all, exist in the SAP Access Control system as well). The feedback from the SAP Access Control is immediate with no requirement for manual steps on the SAP Access Control side. We recommend you to use the centralized provisioning as a default solution.



The centralized provisioning scenario is implemented using the GRC 10.0 or higher Provisioning Framework Version 2 processes AC Validation – Risk Analysis Only (which is recommended) and alternatively AC Validation.

→ Recommendation

Although using AC Validation - Risk Analysis Only is the recommended (simpler) way to implement the centralized provisioning scenario, note the following:

- As this process executes a synchronous call to the GRC system and obtains immediate result – either approved or declined, it does not wait for the result of other processes, such as risk mitigation, that might be triggered in GRC. In this case, the result of the risk mitigation will not be synchronized with SAP Identity Management.
- If you want to configure your SAP Identity Management to wait for the result of a process triggered in GRC, we recommend that you use the Polling scenario.

Note

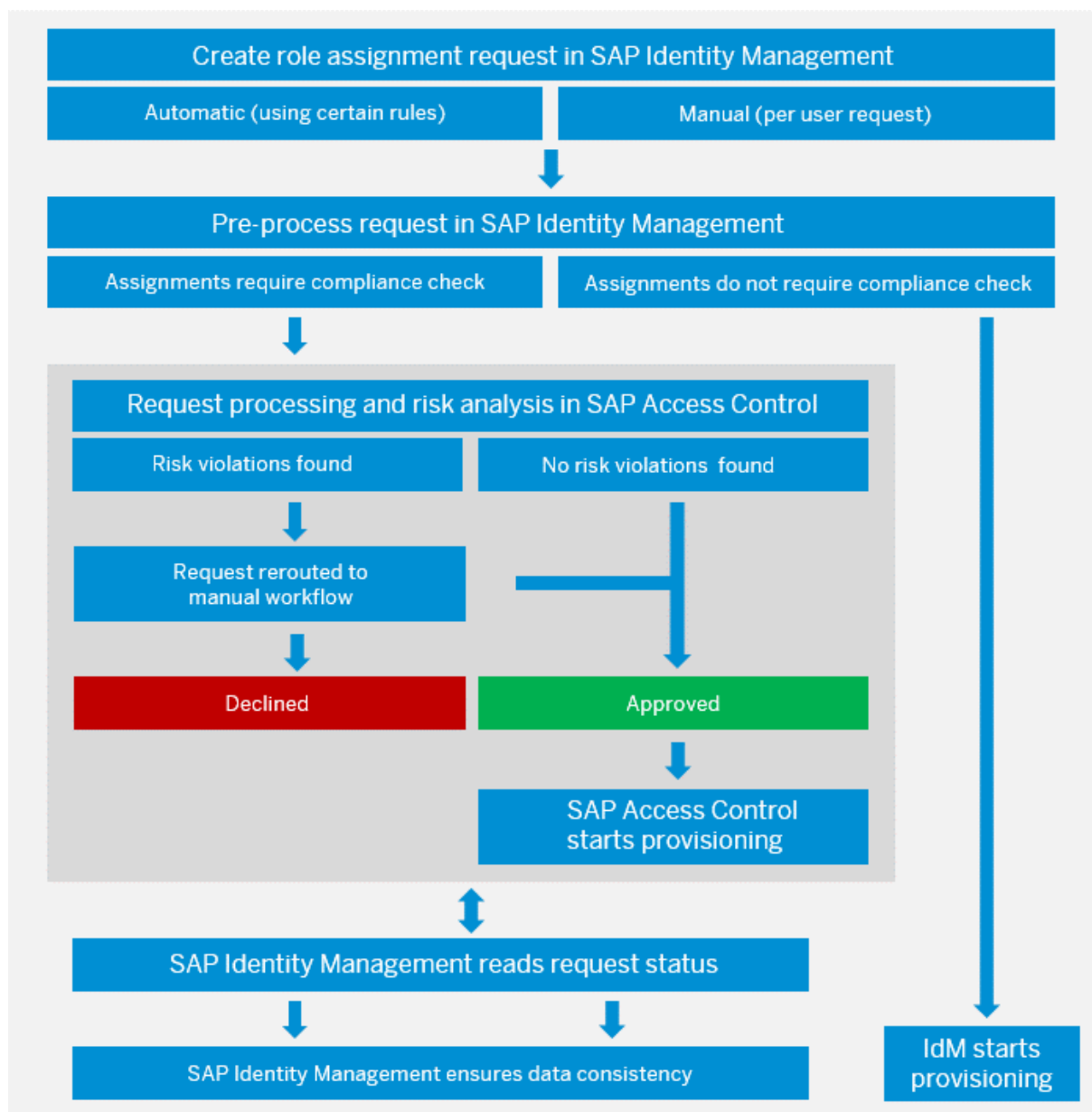
When AC Validation – Risk Analysis Only is used, only the risk analysis result is taken into account. It is immediately returned in SAP Identity Management and the assignment is treated accordingly only based on this result. The process in GRC may continue with the [Request rerouted to manual workflow](#) step (in the figure above), but its result is not taken into account in SAP Identity Management.

Distributed Provisioning

Note

We recommend you to use this solution in exceptional cases only.

This is a scenario where the provisioning is performed both by SAP Identity Management and SAP Access Control. SAP Identity Management is responsible for provisioning the assignments not requiring compliance checks to multiple target systems (both SAP and non-SAP), while SAP Access Control is used for provisioning assignments requiring compliance checks to SAP ABAP target systems (usually a request to SAP Access Control that may require manual approval). This means that no ABAP repositories (and the associated privileges) exist in the Identity Management.



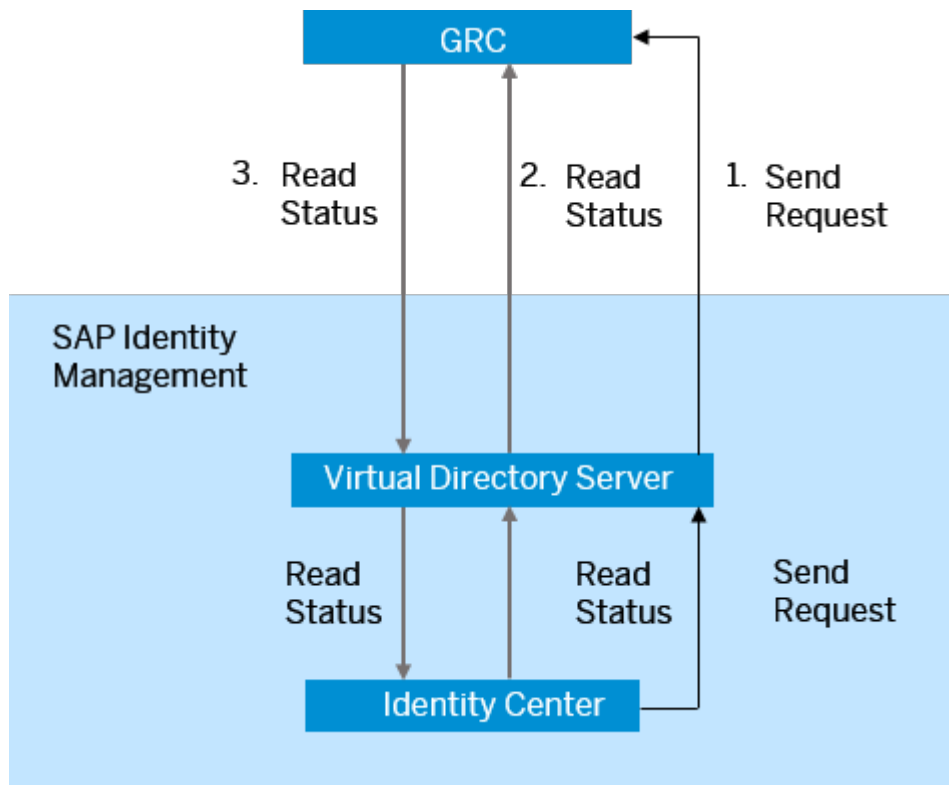
The distributed provisioning scenario is always implemented using the GRC 10.0 or higher Provisioning Framework Version 2 process AC Validation.

6.1.2 Result Handling Scenarios

Whenever a request to SAP Access Control is sent by the SAP Identity Management, further action depends on the results of SAP Access Control's request processing, i.e. which privileges are approved and which are not. Two different approaches to handling a request-processing result exist and only one of them can be active (implemented) at a time.

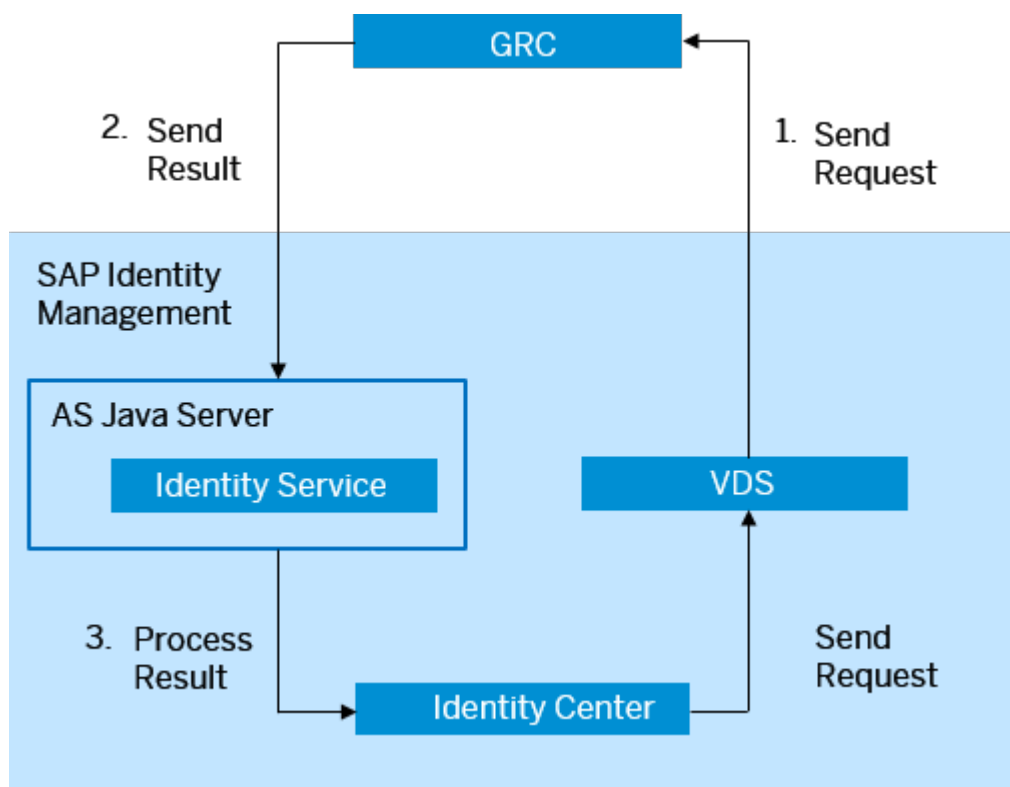
Polling

The Identity Management performs the appropriate web service request, polling the SAP Access Control for the result continuously at a particular interval (which is configurable). This result handling scenario is fail-safe.



Event Based (AC Callback Service)

Instead of polling for the result, the Identity Management is informed about the status of the request when the processing is done. The information about this is sent by GRC (SAP Access Control) by executing its Exit Web Service (GRAC_EXIT_FROM_IDM_WS) call to Identity Services, and the Identity Management is waiting for this signal. Only one signal is sent, which makes this result handling scenario more vulnerable (not fail-safe) as this one signal may get lost (e.g. due to network issues).



The GRC 10.0 or higher Provisioning Framework Version 2 contains processes/workflow to deal with both scenarios (processes AC Polling and AC Callback Service). The behavior of the provisioning framework is controlled by a configurable set of constants on the GRC10 repository.

Using the process AC Validation – Risk Analysis Only, the process executes a synchronous call to the GRC system and obtains the risk analysis result. No use of result handling scenarios polling and callback service is needed. It is when using the process AC Validation that you have to decide which result handling scenario to use and configure.

6.1.3 Provisioning Initiation Scenarios

There are two provisioning initiation scenarios, depending on which system initiates the provisioning.

Note

You can only use one of the two provisioning initiation scenarios.

GRC Initiated Provisioning

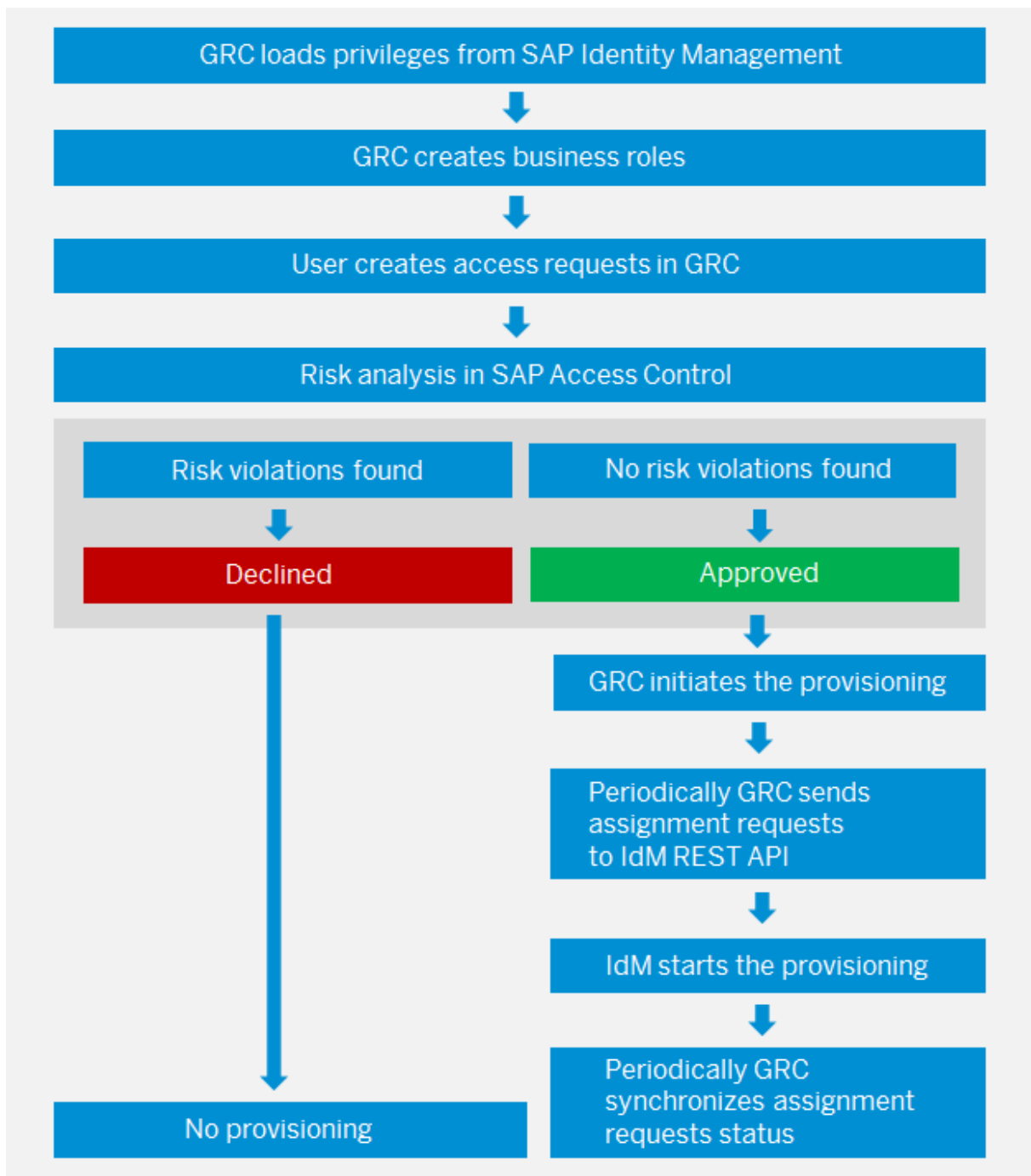
In this scenario, GRC initiates the access request and the provisioning to target systems, while SAP Identity Management is the provisioning system.

Users can make role or privilege assignments in the GRC system, which then initiates the provisioning of those assignments to various target systems through SAP Identity Management. While GRC can provision user assignments to ABAP systems directly, with this scenario it is now possible for GRC to initiate provisioning to SAP HANA systems, AS Java systems and others by using a number of connectors for SAP and non-SAP systems provided with SAP Identity Management.

In addition, this scenario supports creating, modifying and deleting users in target systems through SAP Identity Management.

Before using the GRC initiated provisioning scenario, you need the following prerequisites:

- You have configured the business role synchronization between GRC and SAP Identity Management, so that both systems use a shared role model.
- The GRC administrator user needs the UME role `idm.grc.provisioning` that contains the UME action `idm_bypass_ac_validation`. This is in addition to the following UME actions required by SAP Identity Management REST Interface Version 2: `sap.com_tc~idm~jmx~ump.idm_authenticated` and `sap.com_tc~idm~jmx~ump.idm_authenticated_restapi`.



The high-level steps for the GRC initiated provisioning includes the following:

1. GRC loads privileges from SAP Identity Management.

ⓘ Note

GRC does not load SAP Identity Management privileges with context (that is, context-based assignments).

2. GRC uses the loaded privileges and creates business roles.

3. User creates access requests in GRC.
4. SAP Access Control performs compliance check on the assignments.
5. If no risk violations are found, GRC initiates the provisioning.
6. GRC sends periodically the assignment requests to SAP Identity Management REST API.
7. SAP Identity Management starts the provisioning to the target systems.
8. GRC periodically synchronizes the assignment requests status with SAP Identity Management.

For more information, see [Configuring GRC Initiated Provisioning \[page 729\]](#).

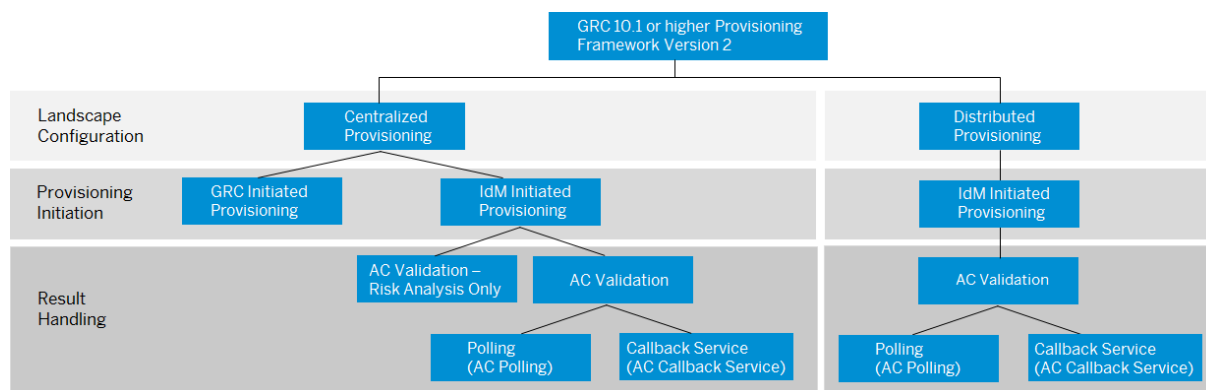
SAP Identity Management Initiated Provisioning

In this scenario, SAP Identity Management initiates the provisioning and it is the the only provisioning system.

For more information, see [Configuring Identity Management Initiated Provisioning \[page 730\]](#).

6.1.4 Integration Scenario Combinations

There are several combinations of integration scenarios that are possible for the integration. See the figure below.



There are four combinations for the centralized provisioning scenario:

- GRC initiated provisioning while using business role synchronization between GRC and SAP Identity Management, so that both systems use a shared role model.
- SAP Identity Management initiated provisioning triggered after using the process AC Validation Risk Analysis Only (no polling or callback service is used). We recommend you to use this way to implement the centralized provisioning scenario.
- SAP Identity Management initiated provisioning triggered after using the process AC Validation with polling as a result handling scenario.
- SAP Identity Management initiated provisioning triggered after using the process AC Validation with callback service as a result handling scenario.

There are two combinations for the distributed provisioning scenario (using the process AC Validation):

- Distributed provisioning with polling.
- Distributed provisioning with callback service.

6.2 Business Role Synchronization

The enhanced integration between SAP Identity Management 8.0 and SAP Access Control (for GRC solutions) 12.0 or higher allows you to synchronize business roles between both systems before user provisioning.

Overview

When business roles are synchronized, SAP Identity Management and GRC use a shared role model. Both systems synchronize the content of the business role (privileges/technical roles). After that only one request for segregation of duties check is needed for the whole content of the role.

Process

GRC is used as the leading system here. First, it reads all relevant privileges from SAP Identity Management through REST API version 2 and uses these privileges to create business roles.

Note

GRC does not read SAP Identity Management privileges that provide specific access to SAP Identity Management User Interface tabs, such as MX_PRIV:WD:TAB_MANAGE.

Second, the business roles and their assigned privileges are loaded back in SAP Identity Management with the Initial Load - Business Roles job.

When the business roles are created or modified, the GRC system calls REST API v2 `SyncGRCRoles` endpoint to notify SAP Identity Management about the change. This endpoint accepts one mandatory parameter `RepositoryName`. Once the notification is received, SAP Identity Management triggers the initial load job for the GRC repository (based on the `RepositoryName` parameter), so that only new or modified roles are loaded.

Creation of Business Roles

Business roles maintained in GRC are created for specific provisioning environment, that is [Development](#), [Testing](#), [Production](#) or [All](#). If a business role in GRC is created for [All](#) environments, SAP Identity Management will load three roles, each of them containing the privileges for the corresponding provisioning environment.

Only privileges belonging to systems in the development, test or production environment are assigned to the user. The environment is appended to the name of the role. For example:

ROLE:GRC:<role name>_PRD

ROLE:GRC:<role name>_TST

ROLE:GRC:<role name>_DEV

All business roles maintained in GRC have the following three attributes defined on them:

Attribute	Description
MX_AC_MAINTAINED	Indicates that a role is created and maintained in the GRC system
MX_AC_ENVIRONMENT	Specifies the GRC provisioning environment
MX_AC_ROLEID	Holds the role name as it is retrieved from the GRC system

Note

All business roles maintained in GRC contain specific attributes that are locked in SAP Identity Management to avoid desynchronization between the two systems. You are not allowed to modify the following attributes: DISPLAYNAME, DESCRIPTION, MX_AC_ENVIRONMENT, MX_AC_ROLEID and MXMEMBER_MX_PRIVILEGE.

Related Information

[Configuring Business Role Synchronization \[page 707\]](#)

[GRC Notification \[page 963\]](#)

6.2.1 Configuring Business Role Synchronization

Prerequisites

- You are using SAP Identity Management 8.0 SP07 or higher and SAP Access Control (for SAP GRC Solutions) 12.0 SP05 or higher.
- You have imported the latest version of `com.sap.idm.grc.grc10` and `com.sap.idm.forms.default` packages.
- You have configured your Virtual Directory Server to access SAP Access Control.
- You have run the initial load jobs for the systems in your SAP Identity Management landscape.

- You have run the initial load service jobs **AC 10.0 – Initial Load – Commons** and **AC 10.0 – Initial Load – Centralized Provisioning**
- You have configured SAP Access Control for business role synchronization with SAP Identity Management, as described here: [Integrating Business Roles with SAP Identity Management and SAP Access Control](#)
- Make sure that either the name of the ABAP repository is the same as the name of the system/application in SAP Access Control, or that the ABAP repository has a repository constant **MX_AC_APPLICATION_ID** which should contain the exact name of the corresponding system/application in the SAP Access Control.

Context

Configuring the business role synchronization scenario involves the following steps:

Procedure

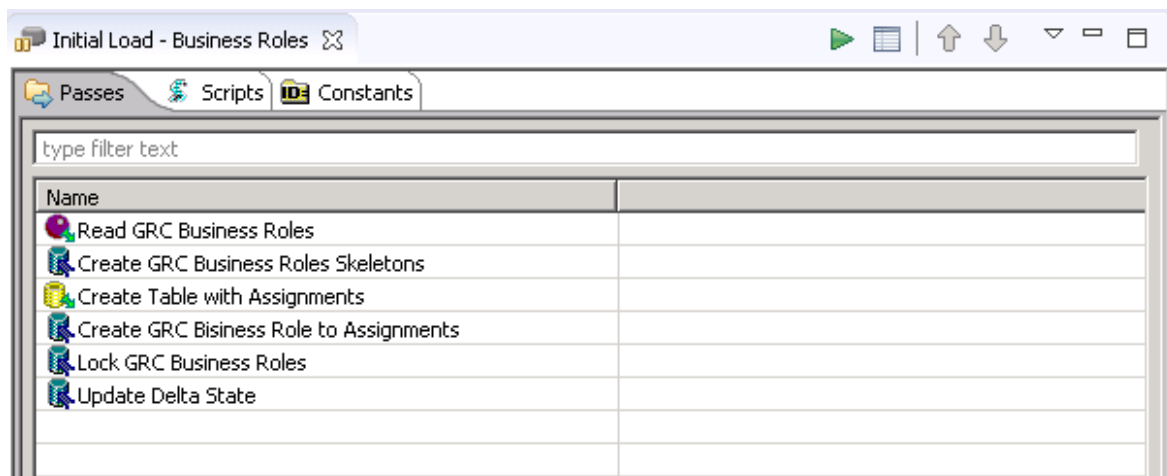
1. The GRC system reads all relevant privileges from SAP Identity Management.

Make sure the **MX_PRIVILEGE_Search** form is selected in the *Search Form* field on the *General* tab of the **MX_PRIVILEGE** entry type. This search form supports the **MX_REPOSITORYNAME** and **MXMEMBER_MX_PERSON** attributes. It allows the GRC system to read all relevant privileges for the specific repository and to check the currently assigned users for each privilege.

2. Create a repository of type **GRC10** with the correct values for the repository constants. See [GRC Related Repository Constants \[page 774\]](#)
3. Activate polling, that is, set the **MX_AC_POLLING_ENABLED** repository constant to 1.
4. Run the **Initial Load – Business Roles** job.

This job tries to load all business roles maintained in GRC and map their privileges (technical roles). Depending on the GRC configuration (defined connection types for GRC connectors and others), some privileges may not be mapped to the business roles, and therefore not loaded in SAP Identity Management.

The **Initial Load – Business Roles** job contains the following passes:



Name	
Read GRC Business Roles	
Create GRC Business Roles Skeletons	
Create Table with Assignments	
Create GRC Business Role to Assignments	
Lock GRC Business Roles	
Update Delta State	

Pass	Description
Read GRC Business Roles	A FromRFC pass that reads GRC data and stores it into two temporary database tables. This pass is configured through the standard JCo configuration parameters needed to establish the connection with the target ABAP system.
Create GRC Business Roles Skeletons	A To Identity store pass that reads the business roles maintained in GRC from a temporary database table and creates the skeletons of the roles (without the assigned privileges) into the identity store.
Create Table with Assignments	A From Database pass that reads all assigned privileges to business roles maintained in GRC from a temporary database table and creates temporary table <code>roleAssignments</code>, which contains GRC role unique identifier and all assignments. The business roles maintained in GRC are stored in unlocked state.
Create GRC Business Role to Assignments	A To Identity store pass that reads all business roles maintained in GRC from a temporary database table and creates the hierarchy of roles and their child assignments into the identity store.
Lock GRC Business Roles	A To Identity store pass that locks the business roles maintained in GRC by setting the <code>MX_AC_MAINTAINED</code> attribute on the roles.
Update Delta State	A To Identity store pass that updates the timestamp of the initial load job. It ensures that only modified data is loaded into the identity store after the initial load.

Note

When SAP Identity Management loads all business roles with mapped privileges or when a Repository Object Synchronization is run to load the back-end privileges into GRC, depending on the size of the load, you may need to increase the value of *Max Search Result Size* property on the *General* tab of the *Identity Store* node in SAP Identity Management Developer Studio. Since SAP Identity Management User Interface SP 08 Patch 16 and SAP Identity Management REST API SP 08 Patch 6, the configured *Max Search Result Size* will no longer be used and all available entries will be loaded. For more information, see [Synchronization Jobs](#).

Next Steps

When the business roles between SAP Identity Management and GRC are synchronized, you can implement one of the following supported scenarios:

- [Configuring GRC Initiated Provisioning \[page 729\]](#)
- [Configuring Centralized Provisioning with Polling \[page 734\]](#)

Note

Using business roles with other scenarios, like Risk Analysis Only and Callback Service, has not been thoroughly tested under all conditions. SAP, therefore, cannot guarantee or imply reliability or function of these scenarios.

6.3 Configuring the Virtual Directory Server

Context

To configure the Virtual Directory Server, do the following:

Procedure

1. In the Virtual Directory Server console, add the Virtual Directory Server configuration, using a provided template.
2. Configure the necessary constants, making a connection to the GRC system.
3. Start the server configuration.

Make sure that it is running with no errors, and that the connection to the GRC system is established.

Related Information

[Adding the Virtual Directory Server Configuration \[page 711\]](#)

[Modifying the \(Template\) Constants \[page 715\]](#)

[Setting the Access Credentials \[page 718\]](#)

[Starting the Virtual Directory Server \[page 720\]](#)

[Testing the Virtual Directory Server Configuration \[page 720\]](#)

6.3.1 Adding the Virtual Directory Server Configuration

Prerequisites

- You have specified the location of the Java runtime environment, path to `Keys.ini` file, defined classpath/JDBC driver and others, when you have installed the Virtual Directory Server. The settings are defined in [Tools > Options...](#).

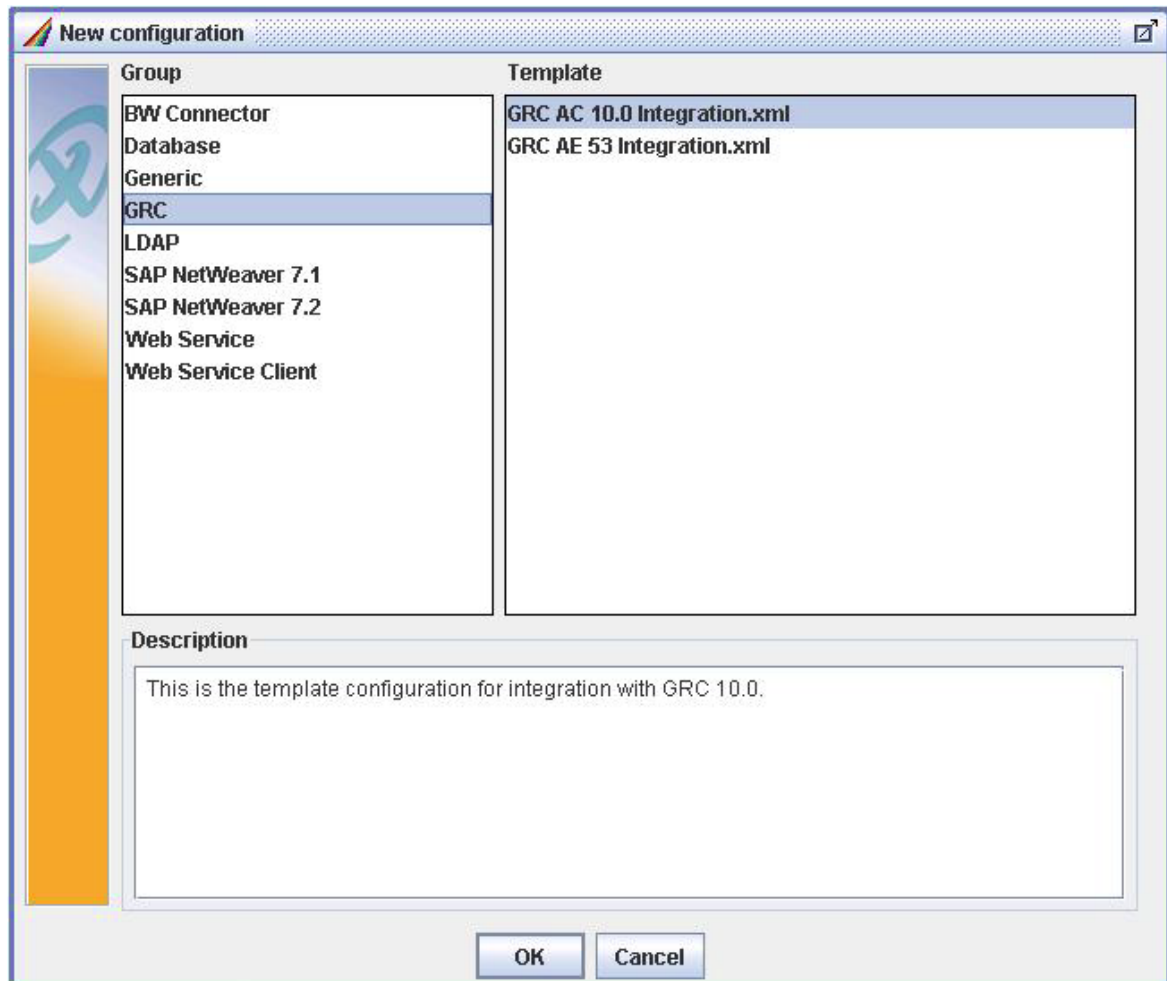
Context

The first step is to create the server configuration in the Virtual Directory Server that the Identity Center uses to access SAP Access Control. The Virtual Directory Server contains a template that can be used to create this configuration.

To create the configuration, do the following:

Procedure

1. In the Virtual Directory Server console, choose [File > New...](#) to open the *New configuration* dialog box.



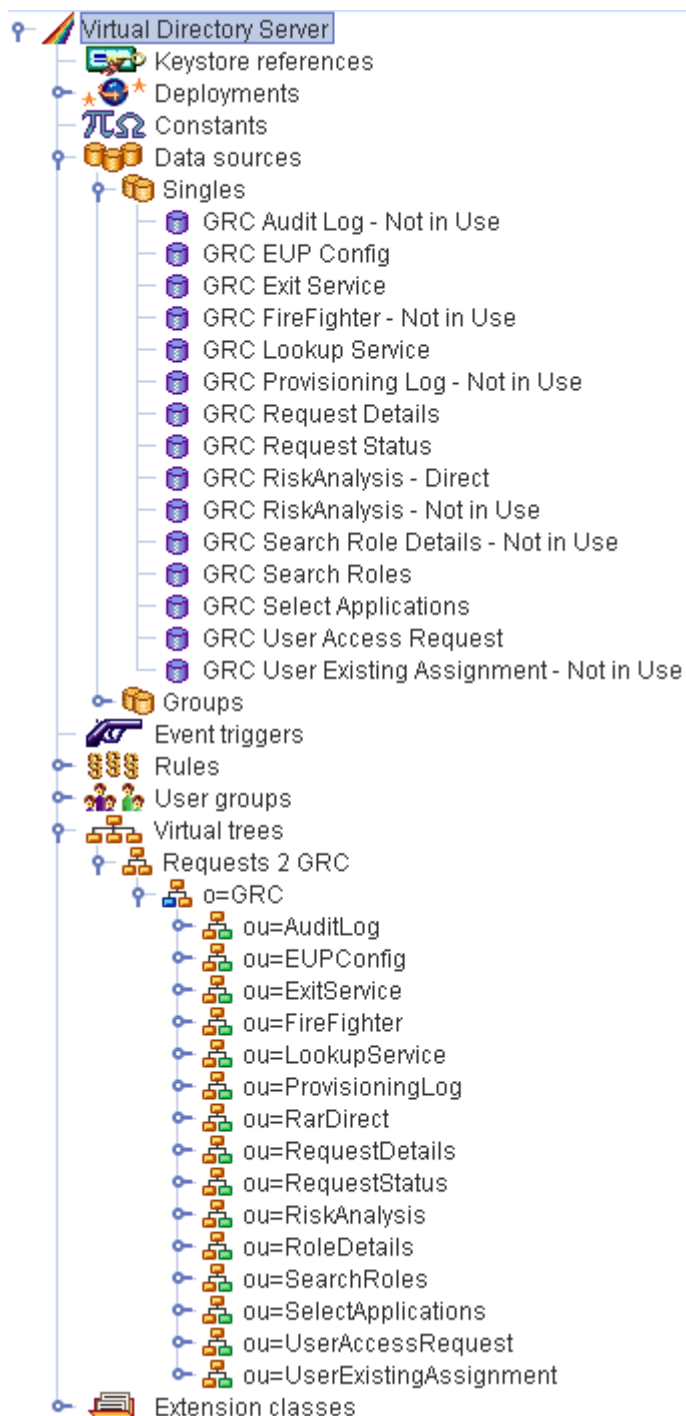
2. In the *Group* list, select *GRC* and in the *Template* list, select *GRC AC 10.0 Integration.xml*.
3. Choose *OK*.

4. Fill in the following values (the names of the corresponding constants created in the Virtual Directory Server configuration are given in parentheses):

Field	Description
<i>Port</i>	Enter the port number that will be used for Virtual Directory Server (when deployed as LDAP server). It is recommended to test and verify the configuration (especially if additional tailoring of the template is done) using an LDAP client, before using it together with the Identity Center.
<i>Web Service URL (URLPREFIX)</i>	The URL to SAP Access Control system, typically <code>http://<server>:<port>/</code>.
<i>Client (URLCLIENT)</i>	Specify the client for the web service.
<i>GRC User and GRC Password (GRCUSER and GRCPWD)</i>	Credentials of the user with access rights to execute web service calls against SAP Access Control.
<i>Identity Center URL (CONNECTION_STRING)</i>	Enter the connection string to the Identity Center database. We recommend that you use the JDBC URL wizard. It is the <code><prefix>_rt</code> user in the Identity Center database that should be used.

5. Choose [OK](#).
6. Enter a file name of the new configuration (e.g. grcintegration.xml) and then choose [Save](#).

The expanded virtual tree looks like this:



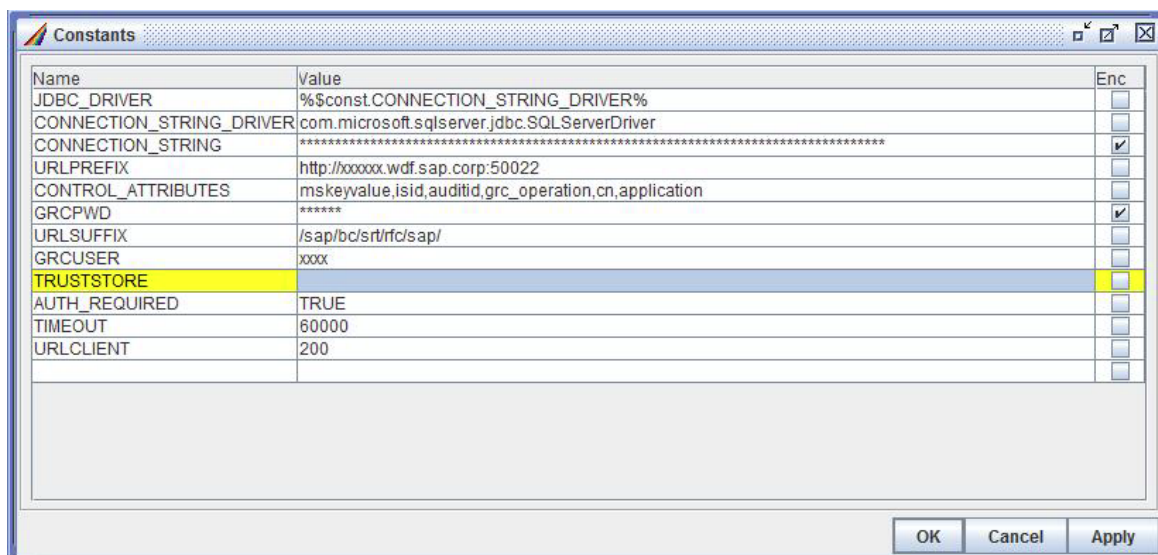
6.3.2 Modifying the (Template) Constants

Context

You can modify the constants entered in the GRC AC 10.0 Integration Template, such as URLPREFIX, GRCUSER, GRCPWD and CONNECTION_STRING, in addition to few other constants. To do so, do the following:

Procedure

1. In the Virtual Directory Server console tree, select [Constants](#) and enter the changes for the constants.



2. Choose [Apply](#).
3. To close the [Constants](#) dialog box, choose [OK](#).

Configuring of the constant TRUSTSTORE is only needed when the GRC server is using the HTTPS communication (SSL/TLS), and requires a keystore to be created first.

6.3.2.1 Configuring the Constant TRUSTSTORE (Optional)

Configuring the TRUSTSTORE constant is optional, and only needed when the GRC server is using the HTTPS communication instead of HTTP, i.e. the server is using SSL/TLS.

To connect your Virtual Directory Server to a GRC system communicating with HTTPS, you need to create a keystore based on either the certificate that was used to sign the GRC server certificate or the certificate used by GRC server directly. Then the TRUSTSTORE constant should point to this keystore file (full path).

Related Information

[Creating a Keystore \[page 716\]](#)

[Defining the Keystore Path in TRUSTSTORE Constant \[page 718\]](#)

6.3.2.1.1 Creating a Keystore

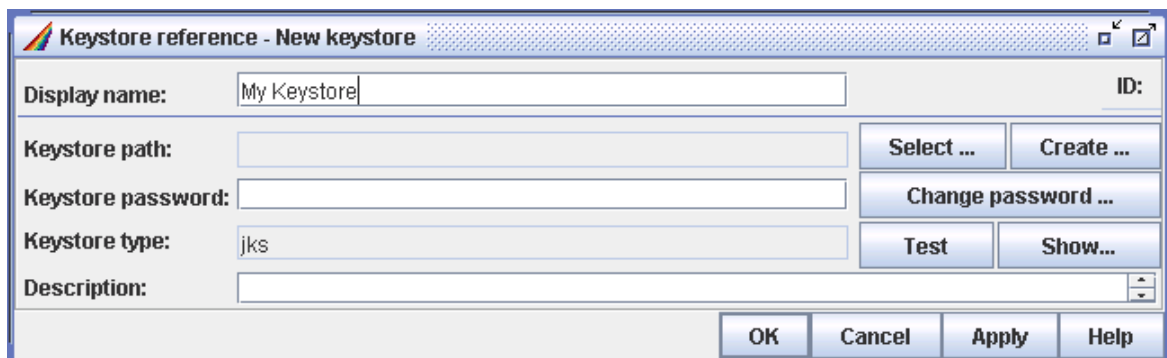
Context

To create a keystore, do the following:

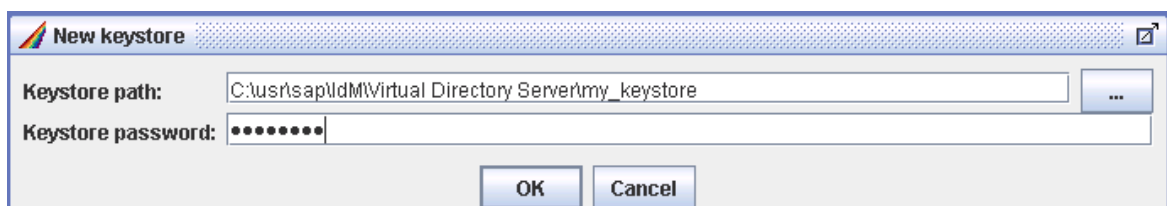
Procedure

1. Export the certificate and make sure that it is available for the Virtual Directory Server. The certificate can be obtained by contacting those responsible for the GRC systems.
2. In the Virtual Directory Server console, select the [Keystore references](#) in the virtual tree of your Virtual Directory Server configuration and select [New...](#) from the context menu.

A dialog box for creating of new keystore reference appears:



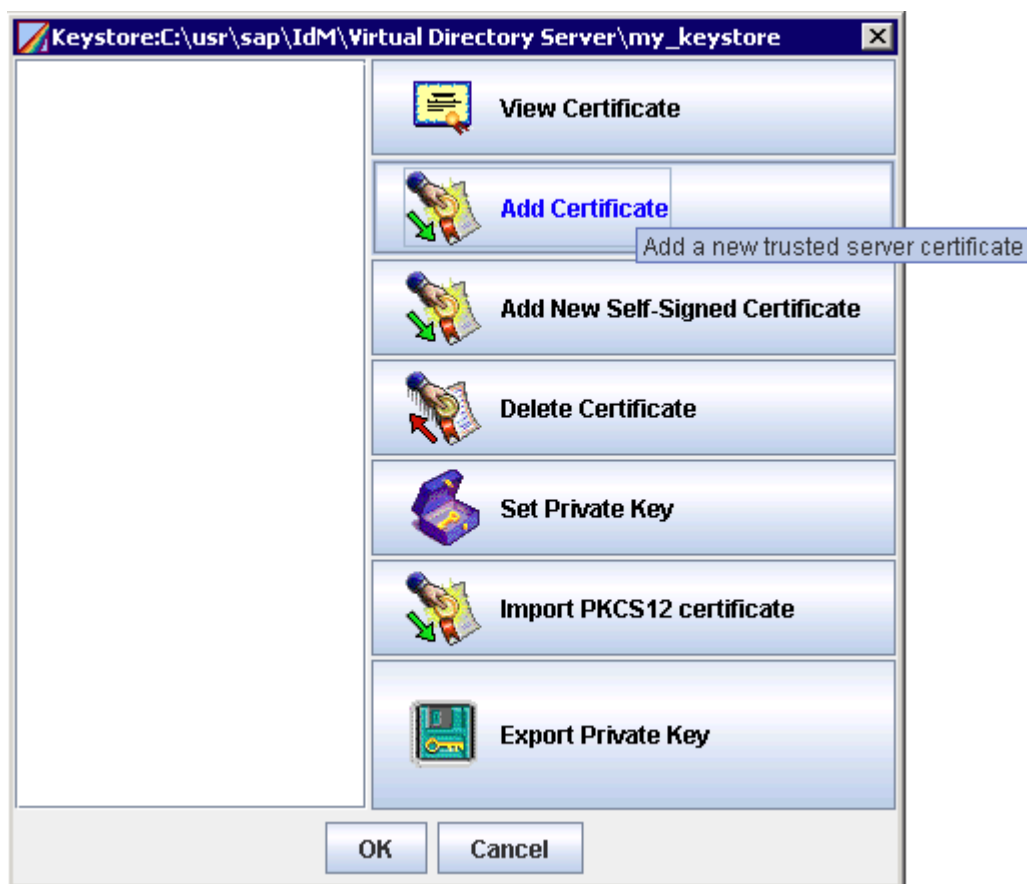
3. Provide the display name for the keystore reference.
4. Choose [Create...](#). A new dialog box appears.



5. Enter a file name and a path to the keystore file you are about to create (alternatively choose ... to navigate to the correct folder and add a file name).

6. Enter the password for the keystore and choose **OK** to save the data and close the dialog box.
7. Choose **Show...**

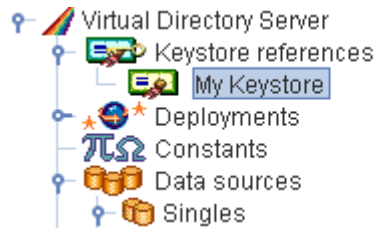
The following dialog box with a set of menus appears.



8. Choose **Add Certificate**, then locate and add the certificate provided previously from the GRC administrator(s).
9. Choose **OK**.
10. Enter the alias for the certificate and choose **OK**. The certificate will be listed in the keystore, in the left panel:



11. Choose **OK** to save the data and close the keystore dialog box.
12. Choose **Apply** and then **OK**. The keystore reference will be added to the virtual tree in your configuration:



6.3.2.1.2 Defining the Keystore Path in TRUSTSTORE Constant

Context

When the keystore is created, the constant TRUSTSTORE needs to hold the path to the created keystore.

You need to do the following:

Procedure

1. Select the [Constants](#) node in the Virtual Directory Server console tree and view the properties.
2. Add the keystore path as the value of the constant TRUSTSTORE.
3. Choose [Apply](#) and then choose [OK](#).

6.3.3 Setting the Access Credentials

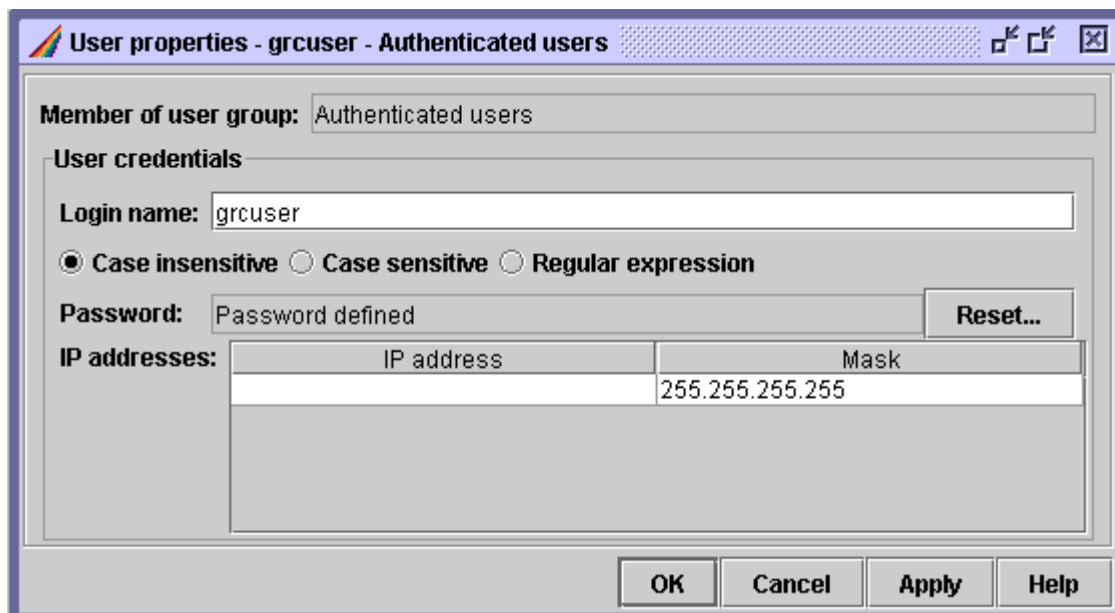
Context

The user access credentials to log on to the Virtual Directory Server are set to grcuser (user)/grcuser (password) as default in the template. We recommend that you do not change these, although you can change them in the configuration.

To change the user name and the password, do the following:

Procedure

1. In the node *User groups* in the console tree, select the default authenticated user *grcuser* and view its properties.



The dialog box is titled "User properties - grcuser - Authenticated users". It contains the following fields and controls:

- Member of user group:** A text box containing "Authenticated users".
- User credentials:**
 - Login name:** A text box containing "grcuser".
 - Case insensitive:** A radio button that is selected.
 - Case sensitive:** An unselected radio button.
 - Regular expression:** An unselected radio button.
 - Password:** A text box containing "Password defined". To its right is a "Reset..." button.
 - IP addresses:** A table with two columns: "IP address" and "Mask". The first row contains "255.255.255.255" in the "Mask" column.

At the bottom of the dialog are four buttons: "OK", "Cancel", "Apply", and "Help".

2. To change the user name, enter the new login name in the *Login name* field.
3. To change the password, choose *Reset...*



The dialog box is titled "Change password". It contains the following fields and controls:

- New password:** A text box with masked characters (asterisks).
- Confirm password:** A text box with masked characters (asterisks).

At the bottom of the dialog are two buttons: "OK" and "Cancel".

4. Enter and confirm the new password.
5. Choose *OK* to close the *Change password* dialog box.
6. Choose *Apply* to apply the changes, and then *OK* to close the *User properties* dialog box.

6.3.4 Starting the Virtual Directory Server

Context

To verify that the server starts without errors, do the following:

Procedure

1. To display the operation log, choose the [Operation](#) button in the menu.
2. Start the server.

If the runtime environment is correct, the Virtual Directory Server will start listening on the configured port.

3. Verify that the server starts in the operation log.

Some typical errors:

- The database driver for your Identity Center database is not in the class path for the Virtual Directory Server. See the help file for the Virtual Directory Server for information about how to extend/configure the class path.
- The selected port is occupied by another process. You can change it by viewing the properties of Deployments/LDAP Deployments/Main listener.

6.3.5 Testing the Virtual Directory Server Configuration

Context

When the server has started successfully, the configuration can be tested using an LDAP client, e.g. LDP (a tool available freely on the internet).

Related Information

[Logging with LDP \[page 721\]](#)

[Performing a Search \[page 722\]](#)

6.3.5.1 Logging with LDP

Prerequisites

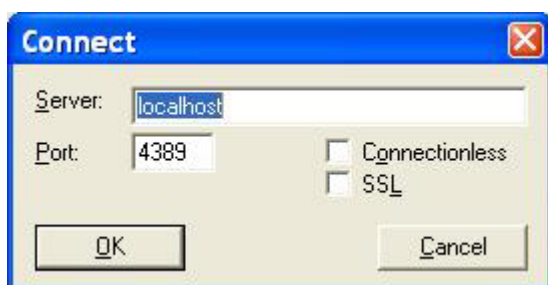
- You have the LDP tool installed. The tool is available freely on the internet.

Context

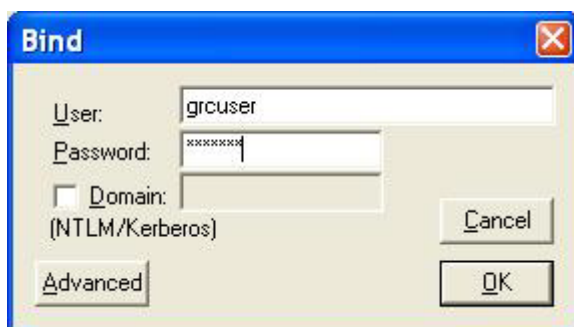
To start logging with LDP, do the following:

Procedure

1. Start LDP.
2. Choose ► *Connection* ► *Connect...* ►.



3. Enter the host name/IP number and port number you specified when configuring the Virtual Directory Server.
4. Choose *OK*.
5. Choose ► *Connection* ► *Bind...* ►.



6. To log on to the Virtual Directory Server, enter **grcuser** (user) and **grcuser** (password) as user credentials. These are the default credentials in the template, but they can be changed in the configuration. For more information, see *Setting the Access Credentials*.

7. Choose [OK](#).

Related Information

[Setting the Access Credentials](#) [page 718]

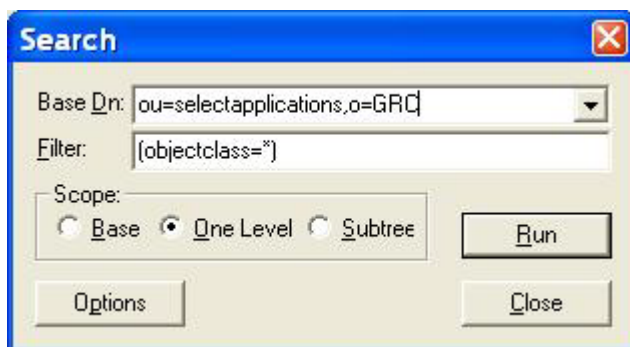
6.3.5.2 Performing a Search

Context

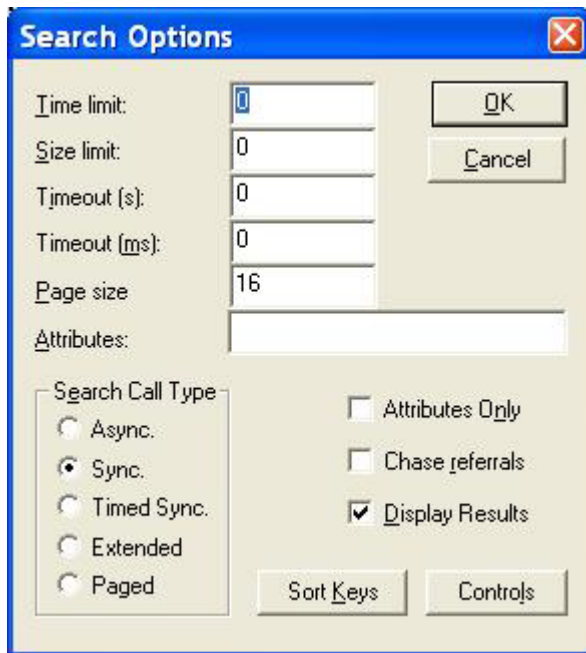
To test the connectivity, perform a search to list the applications in your target SAP Access Control system. Use the DN as shown below, in the field [Base Dn](#).

Procedure

1. Choose [Browse](#) > [Search](#).



2. Choose [Options](#).

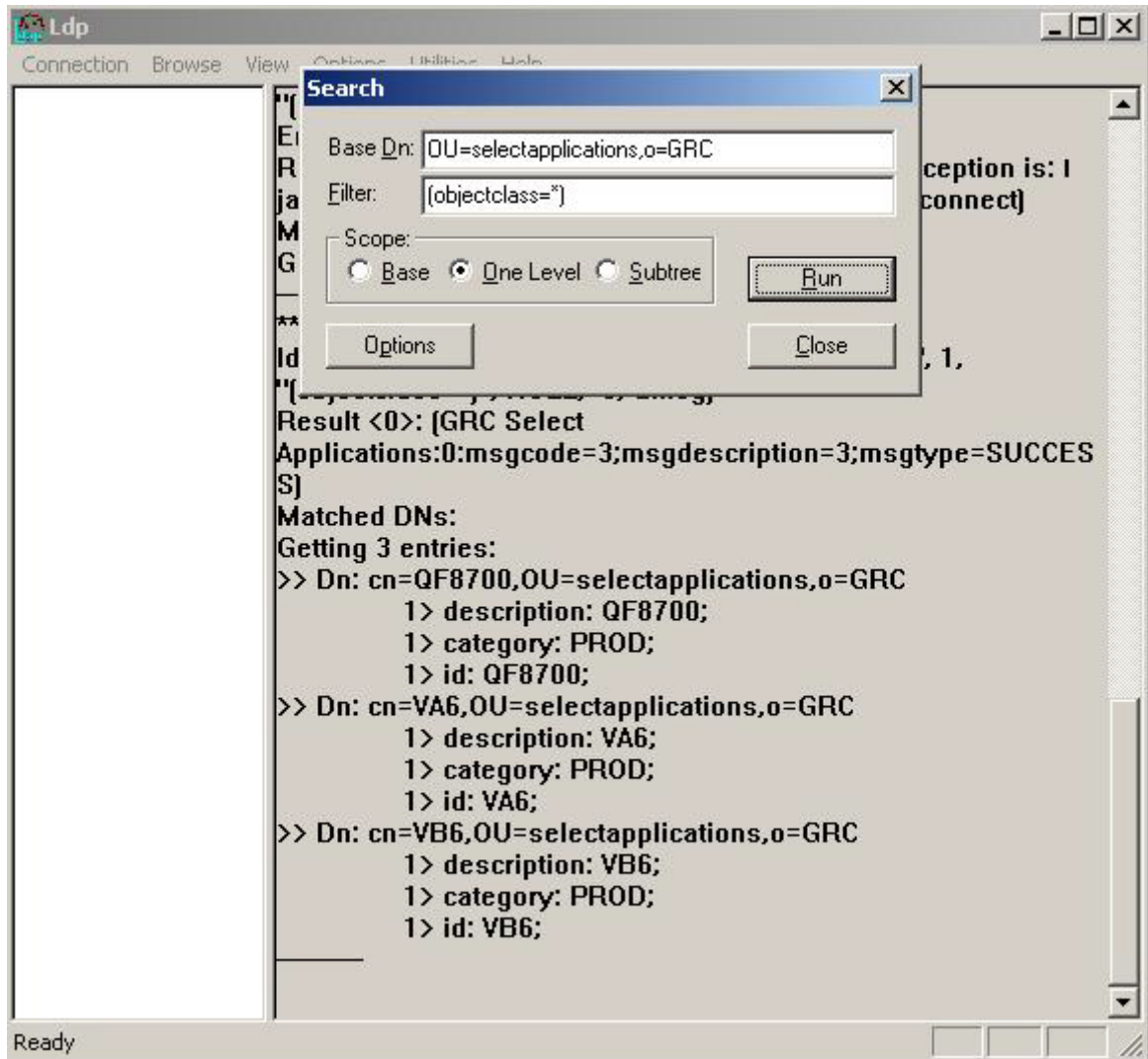


The **Search Options** dialog box contains the following fields and controls:

- Time limit:** Input field with value 0.
- Size limit:** Input field with value 0.
- Timeout (s):** Input field with value 0.
- Timeout (ms):** Input field with value 0.
- Page size:** Input field with value 16.
- Attributes:** Empty text input field.
- Search Call Type:**
 - ☐ Async.
 - ☒ Sync.
 - ☐ Timed Sync.
 - ☐ Extended
 - ☐ Paged
- ☐ Attributes Only
- ☐ Chase referrals
- ☒ Display Results
- Sort Keys:** Button
- Controls:** Button
- OK:** Button
- Cancel:** Button

Make sure that the *Attributes* field is empty.

3. To close the *Search Options* dialog box, choose *OK*.
4. To perform the search, choose *Run*.



The applications returned by the search may vary, depending on what is available in SAP Access Control you are connecting to.

6.4 Importing the GRC Provisioning Framework

Importing the GRC provisioning framework in the Identity Center and configuring the solution makes it possible to submit requests to SAP Access Control from a provisioning solution implemented in the Identity Center.

The GRC provisioning framework is delivered as a separate package [com.sap.idm.grc.grc10](#) in SAP Identity Management 8.0. For details about the package content, see: *GRC Provisioning Framework Package*.

To import the GRC provisioning framework, see *Importing the Provisioning Framework for SAP Identity Management 8.0*.

The package file `com.sap.idm.grc.grc10.idmpck` is located in the folder:
<Install_folder>\Core\ConfigurationPackages\GRC Provisioning Framework.

Related Information

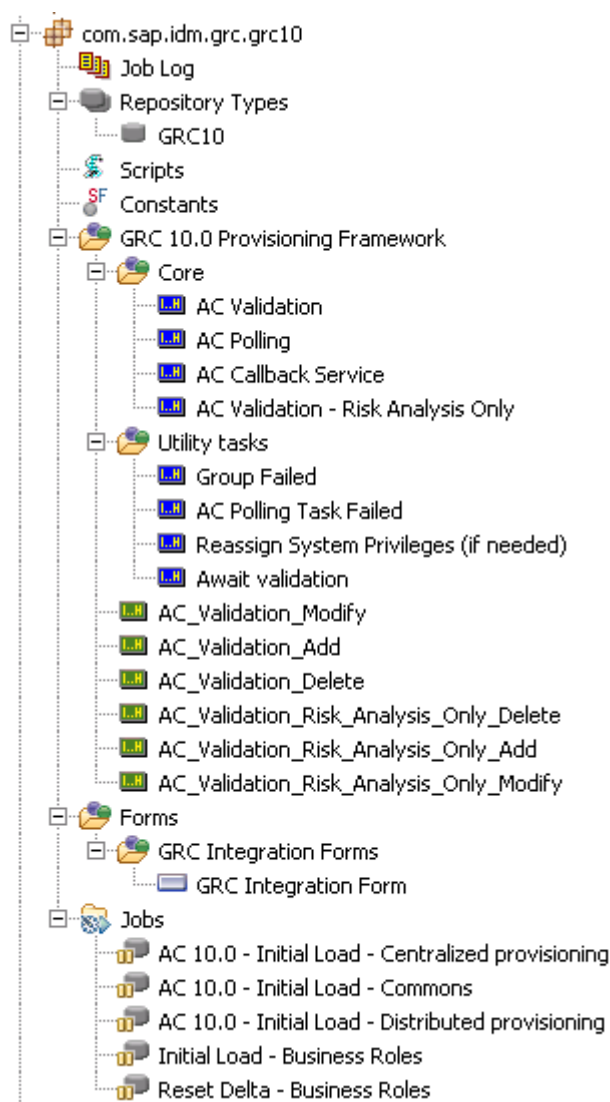
[Importing the Provisioning Framework for SAP Identity Management 8.0 \[page 487\]](#)

[GRC Provisioning Framework Package \[page 725\]](#)

6.4.1 GRC Provisioning Framework Package

The Governance, Risk and Compliance (GRC) Provisioning framework package [com.sap.idm.grc.grc10](#) contains the GRC 10.0 Provisioning Framework, including the repository type, provisioning processes, utility processes, an integration form, initial load jobs and scripts.

The structure of this package looks like this:



Repository Types

The GRC Provisioning framework package contains the following repository type: GRC10. The repository type defines common parameters for all repositories based on this type.

Processes

This package contains a set of internal and public processes.

Internal Processes

Internal Process	Description
Core/AC Callback Service	If you use AC Callback Service process, the GRC/AC system sends the validation result to the Identity Management through a defined Identity Service on the AS Java (VDS), i.e. the Identity Management is waiting until notified.
Core/AC Polling	If you use AC Polling process to obtain the validation result, the Identity Management is polling for the result as configured in the process.
Core/AC Validation	If you use AC Validation process, the validation result is obtained either through AC Polling process or AC Callback Service process, i.e. the AC Validation process needs to wait until either AC Polling or AC Callback Service has completed.
Core/AC Validation – Risk Analyses Only	If you use AC Validation - Risk Analysis Only process, the validation result is obtained through a synchronous call to GRC/AC system.
Utility tasks/AC Polling Task Failed	Utility task AC Polling Task Failed is executed if the approval results in an error (due to a timeout for example).
Utility tasks/Group Failed	Utility task Group Failed is executed if an error occurs while processing the risk analysis, meaning that something unpredictable has happened, and no status is obtainable.
Utility tasks/Reassign System Privilege (if needed)	The utility task Reassign System Privileges (if needed) is used to recalculate the system privileges that the entry should have.
Utility tasks/Await validation	The utility task Await validation is used to wait for SAP Access Control to process the request (that is, to approve or reject the privilege or role assignment).

Public Processes

Public Process	Description
AC_Validation_Add	
AC_Validation_Delete	
AC_Validation_Modify	
AC_Validation_Risk_Analyses_Only_Add	
AC_Validation_Risk_Analyses_Only_Delete	
AC_Validation_Risk_Analyses_Only_Modify	

Forms

The GRC provisioning framework package provides a [GRC Integration Form](#) for the integration between GRC and SAP Identity Management. It is used by GRC for REST calls to SAP Identity Management to retrieve users and privileges when synchronizing them to SAP Access Control.

The TASK GUID of this form must be added in GRC when configuring the connectors to SAP Identity Management. To get the TASK GUID, first you need to select [Enable "Get GUID" for jobs and processes](#) option in the SAP Identity Management Developer Studio main menu (▮ [Window](#) ▸ [Preferences](#) ▸ [Global Settings](#) ▮). Then, select the form and choose [Get GUID](#) from the context menu.

This form is delivered with the default form's access control, that is Logged-in user or identity store entry with access to the Manage tab, and is not displayed in the SAP Identity Management User Interface.

Jobs

The GRC provisioning framework package provides the following service jobs that are used for the initial load. For more information, see *Running the Initial Load Service Jobs*.

Job	Description
AC 10.0 - Initial Load - Centralized provisioning	Run the job AC 10.0 - Initial Load - Centralized provisioning if deploying the centralized landscape configuration scenario.
AC 10.0 - Initial Load - Commons	Run the job AC 10.0 - Initial Load - Commons disregarding of what provisioning scenario is implemented, centralized or distributed.
AC 10.0 - Initial Load - Distributed provisioning	Run the job AC 10.0 - Initial Load - Distributed provisioning if deploying the distributed landscape configuration scenario.

Job	Description
Initial Load – Business Roles	Run the job Initial Load – Business Roles to retrieve the business roles maintained in GRC and the assigned privileges from the connected GRC system and store them into the identity store. This job supports delta mechanism.
Reset Delta – Business Roles	Run the job Reset Delta – Business Roles to reset the delta information. When the Initial Load – Business Roles job is run, the date of the last job execution is stored in the <LastModificationTimeBusinessRoles> repository variable. The Reset Delta – Business Roles job resets the value of this repository variable, so that a complete load is performed.

Along with the processes, this package contains several scripts used in various cases, such as getting user manager and setting request reason.

Related Information

[Running the Initial Load Service Jobs \[page 750\]](#)

6.5 Configuring the Solution

There are five different integration scenario combinations:

- Centralized provisioning with AC Validation – Risk Analysis Only process
- Centralized provisioning using polling
- Centralized provisioning using callback service
- Distributed provisioning using polling
- Distributed provisioning using callback service

Configuration details for each integration scenario are described in the subsections below.

Note

All configuration setups involving configuration on privileges require the Initial Load service jobs to be executed first, before configuring the privileges. See *Initializing the Solution – Running the Initial Load Service Jobs* section for details. Basing the solution on an extensive configuration on the privileges is not recommended and should be avoided if possible.

6.5.1 Configuring GRC Initiated Provisioning

When you configure your SAP Identity Management 8.0 and SAP Access Control (for GRC solutions) 12.0 integration, you want GRC to initiate the provisioning of user assignments to target systems.

Prerequisites

- Both systems have synchronized their business roles and privileges (technical roles).
- The GRC administrator user needs the UME role `idm.grc.provisioning` that contains the UME action `idm_bypass_ac_validation`. This is in addition to the following UME actions required by SAP Identity Management REST Interface Version 2: `sap.com_tc~idm~jmx~ump.idm_authenticated` and `sap.com_tc~idm~jmx~ump.idm_authenticated_restapi`.

Context

In this scenario, the user creates access request in GRC. When it is approved, GRC initiates the provisioning which is triggered by SAP Identity Management. While GRC can provision user assignments to ABAP systems directly, for non-ABAP systems (such as SAP HANA systems, AS Java systems and others), it needs SAP Identity Management and the connectors in SAP provisioning framework.

To configure the GRC initiated provisioning, proceed as follows:

Procedure

1. Logon to your GRC system and start the **NWBC** transaction.
2. Navigate to ► [Access Management](#) ► [Access Requests Creation](#) ► [Access Request](#) ► and provide the required information in the [Request Details](#) area.
3. On the [User Access](#) tab, choose ► [Add](#) ► [Role](#) ► and search for the business roles maintained in GRC that you want to assign to the user(s).
4. Specify the [Provisioning Environment](#) and the [Provisioning Actions](#), and [Submit](#) the request.
5. GRC performs compliance check on the assignment(s).
6. When the request is approved, provisioning to the target systems is triggered.

Depending on the content of the assigned business role and the GRC configuration (defined connection types for GRC connectors and others), the provisioning is triggered either by SAP Identity Management, or by GRC.

Results

If the assigned business role contains ABAP and non-ABAP privileges (SAP HANA, AS Java and others), the provisioning can be triggered as follows:

- SAP Identity Management provisions all privilege assignments.
- SAP Identity Management provisions only privilege assignments to non-ABAP systems (hybrid scenario)
- GRC provisions privilege assignments to ABAP systems.

GRC periodically synchronizes the assignment requests status with SAP Identity Management.

Note

If a business role is assigned to a user in GRC, and the privilege assignments fail, you can expect the following:

- If all privilege assignments fail, both systems will have the same assignment request status: **Failed**.
- If some privilege assignments succeed and some fail, in GRC the assignment request status is: **Failed**, while in SAP Identity Management, the status of successful assignments is **OK**, and the status of the failed ones is **Failed**.
In this case, you can retry the failed assignments in SAP Identity Management. Once they succeed and their status is **OK**, you need to run the *Repository Object Synch* job in GRC, so that both systems synchronize the assignment requests status.

Note

If a request is canceled in GRC, it gets status **Aborted**. The **Aborted** status is supported as of SAP Identity Management 8.0 SP07 Virtual Directory Server patch level 1.

6.5.2 Configuring Identity Management Initiated Provisioning

When you configure your SAP Identity Management 8.0 and SAP Access Control (for GRC solutions) 12.0 integration, you want SAP Identity Management to initiate the provisioning of user assignments to target systems.

Prerequisites

- Both systems have synchronized their business roles and privileges (technical roles).
- The GRC administrator user needs the UME role `idm.grc.provisioning` that contains the UME action `idm_bypass_ac_validation`. This is in addition to the following UME actions required by SAP Identity Management REST Interface Version 2: `sap.com_tc~idm~jmx~ump.idm_authenticated` and `sap.com_tc~idm~jmx~ump.idm_authenticated_restapi`.

Context

In this scenario, the user creates access request in SAP Identity Management. The request is sent over to GRC for risk analysis and remediation. When it is approved, SAP Identity Management triggers the provisioning.

To configure the Identity Management initiated provisioning, proceed as follows:

Procedure

1. In the Identity Management User Interface, assign a GRC maintained business role to a user.
2. GRC performs compliance check on the assignment(s).
3. When the request is approved, SAP Identity Management triggers the provisioning to the target systems.

6.5.3 Configuring Centralized Provisioning Scenario

Prerequisites

You have configured the [No Provisioning](#) option in your GRC system. See [Configuring No Provisioning in the GRC System \[page 732\]](#)

Overview

The SAP Identity Management is the only provisioning system in this scenario, responsible for provisioning both the assignments that require and do not require compliance checks to the target systems. SAP Identity Management uses SAP Access Control to execute risk analysis.

This means that the validation process (risk analysis) is performed according to the configuration of the GRC10 repository. Depending on which of the sub-scenarios (described in the subsections below) is used, the repository constants MX_VALIDATE_ADD/DEL_TASK on the GRC10 repository are configured to reference either process AC Validation – Risk Analysis Only or AC Validation.

Note

By defining both MX_VALIDATE_ADD/DEL_TASK, you are defining compliance checks for both add and delete operations. You can choose which operations you would like to perform the risk analysis for, and define the validate process for those operations. To run the checks for e.g. the add operation but not for the delete operation, define the MX_VALIDATE_ADD_TASK, but not the MX_VALIDATE_DEL_TASK.

In addition to the GRC10 repository, the Identity Management has at least one ABAP repository and the privileges associated with it (the privileges, some or all, exist in the SAP Access Control system as well). The

provisioning of the privileges is performed according to the configuration of the ABAP repository that the privileges belong to (MX_ADD/DEL_MEMBER_TASK defined on the ABAP repository).

Note

These constants (MX_VALIDATE_ADD/DEL_TASK and MX_ADD/DEL_MEMBER_TASK) may also be configured on privileges (on each privilege individually). The recommended (and simpler) configuration is leaving these constants on privileges to be inherited from the repositories.

During the initial load, the constants on the privileges are inherited from the repository defined by the constant MX_REPOSITORY_VALIDATE (in this case the repository GRC10). This is in addition to the repository they belong to (an ABAP repository defined by the constant MX_REPOSITORYNAME).

Note

All configuration setups involving configuration on privileges require the Initial Load service jobs to be executed first, before configuring the privileges. Basing the solution on an extensive configuration on the privileges is not recommended and should be avoided if possible.

The subsections describe the three integration scenario combinations for the centralized provisioning scenario.

Related Information

[Centralized Provisioning with AC Validation – Risk Analysis Only Task \(Recommended\) \[page 733\]](#)

[Configuring Centralized Provisioning with Polling \[page 734\]](#)

[Configuring Centralized Provisioning with Callback Service \[page 735\]](#)

6.5.3.1 Configuring No Provisioning in the GRC System

You configure the *No Provisioning* option in the GRC system to ensure that SAP Identity Management is the only provisioning system in the centralized provisioning scenario.

Procedure

1. Log on to your GRC system and start the SPRO transaction.
2. Choose *SAP Reference IMG*.
3. In the structure, navigate to  *Governance, Risk and Compliance*  *Access Control*  *User Provisioning*  *Maintain Provisioning Settings* .
4. In the *Maintain System Provisioning Configuration* dialog, select *Auto Provisioning* and choose *No Provisioning*.

6.5.3.2 Centralized Provisioning with AC Validation – Risk Analysis Only Task (Recommended)

Context

To configure this scenario in the Identity Management Administration User Interface, do the following:

On the GRC10 repository

Procedure

1. Define the process reference attributes MX_VALIDATE_ADD_TASK and MX_VALIDATE_DEL_TASK. Use the GRC provisioning framework process AC Validation – Risk Analysis Only as validate process, which is the recommended for the centralized provisioning scenario.
2. (Optional) Configure the privilege grouping for the operations involving the GRC10 repository (i.e. validation process). See *Grouping of Privileges*.

On the ABAP repository

Procedure

1. Define the process reference attributes MX_ADD_MEMBER_TASK and MX_DEL_MEMBER_TASK (For example, reference to SAP Provisioning Framework task(s)).
2. (Optional) Configure the privilege grouping for the operations involving the ABAP repository (i.e. provisioning process). See *Grouping of Privileges*.

Note

Make sure that either the name of the ABAP repository is the same as the name of the system/application in SAP Access Control, or that the ABAP repository has a repository constant MX_AC_APPLICATION_ID which should contain the exact name of the corresponding system/application in the SAP Access Control.

6.5.3.3 Configuring Centralized Provisioning with Polling

Context

To configure centralized provisioning with polling in the Identity Management Administration User Interface, do the following:

On the GRC10 repository

Procedure

1. Define the process reference attributes MX_VALIDATE_ADD_TASK and MX_VALIDATE_DEL_TASK. Use the GRC provisioning framework processes AC_Validation_Add and AC_Validation_Delete.
2. When using the process AC Validation as the validate process, the repository constant MX_AC_FIX_SYSTEM_PRIVILEGES must be set to **0**.

Note

The default value is **1**. You must always set it to **0** when using AC Validation for centralized provisioning scenario. For more information, see [GRC Related Repository Constants \[page 774\]](#).

3. To activate polling, set the repository constant MX_AC_POLLING_ENABLED to **1** and define reference to the process AC Polling in the repository constant MX_AC_POLLING_TASK.
4. (Optional) Configure the privilege grouping for the operations involving the GRC10 repository (i.e. validation process). See *Grouping of Privileges*.

On the ABAP repository

Procedure

1. Define the process reference attributes MX_ADD_MEMBER_TASK and MX_DEL_MEMBER_TASK (e.g. reference to a SAP Provisioning Framework process).
2. (Optional) Configure the privilege grouping for the operations involving the ABAP repository (i.e. provisioning process). See *Grouping of Privileges*.

Note

Make sure that either the name of the ABAP repository is the same as the name of the system/application in SAP Access Control, or that the ABAP repository has a repository constant MX_AC_APPLICATION_ID which should contain the exact name of the corresponding system/application in the SAP Access Control.

6.5.3.4 Configuring Centralized Provisioning with Callback Service

Context

To configure centralized provisioning with callback service in the Identity Management Administration User Interface, do the following:

On the GRC10 repository

Procedure

1. Define the process reference attributes MX_VALIDATE_ADD_TASK and MX_VALIDATE_DEL_TASK. Use the GRC provisioning framework task AC Validation.
2. When using the process AC Validation as the validate process, the repository constant MX_AC_FIX_SYSTEM_PRIVILEGES must be set to 0.
3. To activate callback service, set the repository constant MX_AC_POLLING_ENABLED to 0. In addition, an identity service needs to be provided for the callback service, which is a VDS deployed to your AS Java server. See *Configuring the Identity Services (for Callback Service)* section for more details. For the configuration of the callback service in the GRC system/SAP Access Control, see *Configuring the Callback Service Result Handling Scenario in GRC System/SAP Access Control* section.
4. (Optional) Configure the privilege grouping for the operations involving the GRC10 repository (i.e. validation process). See section *Grouping of Privileges*.

On the ABAP repository

Procedure

1. Define the process reference attributes MX_ADD_MEMBER_TASK and MX_DEL_MEMBER_TASK (e.g. reference to a SAP Provisioning Framework task).
2. (Optional) Configure the privilege grouping for the operations involving the ABAP repository (i.e. provisioning process). See *Grouping of Privileges*.

6.5.4 Configuring Distributed Provisioning Scenario

In the distributed provisioning scenario, no ABAP repositories in addition to the GRC10 repository exist in the Identity Management.

The assignment of the privileges associated with the GRC10 repository (the privileges, some or all, exist in the SAP Access Control system as well) does not require compliance checks. The GRC/SAP Access Control system handles the assignments requiring the compliance checks. The AC Validation process is defined as the MX_ADD/DEL_MEMBER_TASK on the GRC10 repository. MX_VALIDATE_ADD/DEL_TASK on the GRC10 repository are optional (may be e.g. an approval task).

The subsections describe the two integration scenario combinations for the distributed provisioning scenario.

Related Information

[Configuring Distributed Provisioning with Polling \[page 736\]](#)

[Configuring Distributed Provisioning with Callback Service \[page 737\]](#)

6.5.4.1 Configuring Distributed Provisioning with Polling

Context

To configure distributed provisioning with polling in the Identity Management Administration User Interface, do the following on the GRC10 repository:

Procedure

1. Make sure that the repository constant MX_AC_FIX_SYSTEM_PRIVILEGES is set to **1**.
2. Define the task reference attributes MX_ADD_MEMBER_TASK and MX_DEL_MEMBER_TASK. Use the GRC provisioning framework process AC Validation, which is the recommended process for the distributed provisioning scenario.
3. To activate polling, set the repository constant MX_AC_POLLING_ENABLED to **1** and define reference to the process AC Polling in the repository constant MX_AC_POLLING_TASK.
4. (Optional) Configure the privilege grouping. See section *Grouping of Privileges*.

6.5.4.2 Configuring Distributed Provisioning with Callback Service

Context

To configure distributed provisioning with callback service in the Identity Management Administration User Interface, do the following on the GRC10 repository:

Procedure

1. Make sure that the repository constant MX_AC_FIX_SYSTEM_PRIVILEGES is set to **1**.
2. Define the task reference attributes MX_ADD_MEMBER_TASK and MX_DEL_MEMBER_TASK. Use the GRC provisioning framework process AC Validation, which is the recommended process for the distributed provisioning scenario.
3. To activate callback service, set the repository constant MX_AC_POLLING_ENABLED to **0**. In addition, an identity service needs to be provided for the callback service, which is a VDS deployed to your AS Java Server. Details are described in section *Configuring the Identity Services (for Callback Service)*. For the configuration of the callback service in the GRC system/SAP Access Control, see section *Configuring the Callback Service Result Handling Scenario in GRC System/SAP Access Control*.
4. (Optional) Configure the privilege grouping. See section *Grouping of Privileges*.

6.5.5 Configuring the Identity Services (for Callback Service)

When implementing the event-based result handling scenario (AC Callback Service), you need to configure and deploy a Virtual Directory Server configuration on the AS Java as an identity service to communicate with the GRC system.

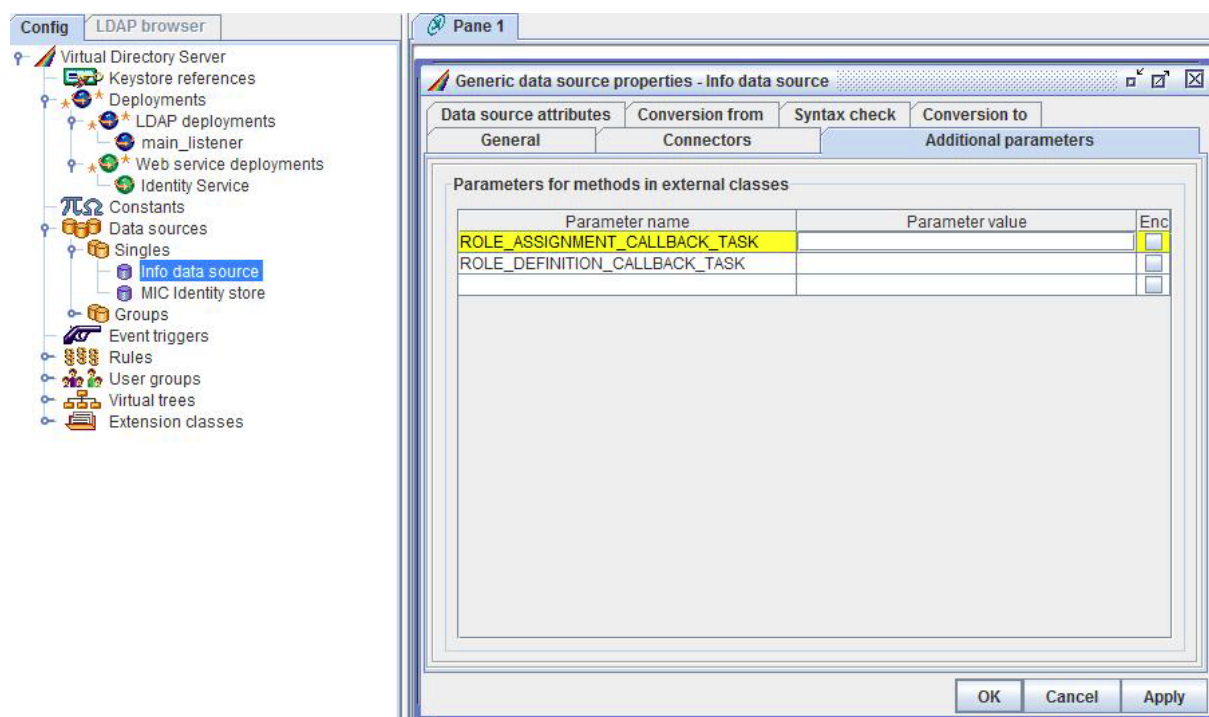
Note

This needs to be configured in a separate Virtual Directory Server configuration (i.e. not in the configuration described in *Configuring the Virtual Directory Server* in this section).

For details about Identity Services, the configuration and deploying of the configuration as a service on the AS Java server, see the document *SAP NetWeaver Identity Management Identity Services Configuration Guide* and SAP NetWeaver Identity Management Identity Services Architectural overview available on SAP Community Network (SCN).

When creating an identity service configuration in the Virtual Directory Server, define the AC Callback Service process in the template parameter Role Assignment (enter the correct task ID), as described in the above mentioned configuration guide. Alternatively, make sure that the following parameters are configured in the

identity service configuration in the Virtual Directory Server (on the *Additional parameters* tab of the *Info data source* properties) before the configuration is deployed as the AS Java service. See the figure below.



Use the same values as specified in the Identity Center:

Constant name	Description
ROLE_ASSIGNMENT_CALLBACK_TASK	Holds the task ID of the process AC Callback Service from the GRC 10.0 Provisioning Framework Version 2.
ROLE_DEFINITION_CALLBACK_TASK	For future use.

Related Information

[SAP NetWeaver Identity Management Identity Services Configuration Guide](#)

[SAP NetWeaver Identity Management Identity Services Architectural overview](#)

6.5.6 Configuring the Callback Service Result Handling Scenario in GRC System/SAP Access Control

Context

To configure the Callback Service on the GRC system/SAP Access Control, you need to complete the following steps:

Procedure

1. Activate GRC Web Services.
2. Create a connector (for the Callback Service result handling scenario).
3. Maintain connectors and connection types.
4. Maintain connection settings.
5. Maintain connector settings in Access Control.
6. Maintain mappings for actions and connector groups.

The first five steps are generic configuration steps for the integration between SAP Identity Management and SAP Access Control, i.e. these steps are defining the connector. Step six is callback service result handling scenario specific.

To perform the steps, do the following:

1. Login to your GRC system and start the SPRO transaction.
2. Choose [SAP Reference IMG](#).
3. In the structure, navigate to [Governance, Risk and Compliance](#).

Related Information

[Activating GRC Web Services \[page 740\]](#)

[Creating a Connector \[page 740\]](#)

[Maintaining Connectors and Connection Types \[page 741\]](#)

[Maintaining Connector Settings in Access Control \[page 742\]](#)

[Maintaining Mapping for Actions and Connector Groups \[page 743\]](#)

6.5.6.1 Activating GRC Web Services

Context

You need to generate the GRC Service Providers to activate the GRC Web Services. This allows SAP Identity Management to access the services via GRC Web Service calls.

To activate the GRC Web Services, do the following:

Procedure

1. In the structure of your GRC system, navigate to Common Component Settings/Integration Framework.
2. Start the IMG activity Maintain Service Providers and Consumer Proxies in SOA Manager.
3. Follow the IMG activity documentation to activate the web services for either all web services or only the one(s) required for your scenario. For more information about the available web services see the document *SAP Access Control 10.0 Interface for Identity Management*.

Both the service name and the binding name should be the same as the web service name. When applying the settings, select *User ID/Password* as Transport Channel Authentication.

6.5.6.2 Creating a Connector

Context

You need to create a connector which can be used for Callback Service and a HTTP connection.

To create a connector, do the following:

Procedure

1. In the structure of your GRC system, navigate to Common Component Settings/Integration Framework.
2. Start the IMG activity *Create Connectors*.
3. Follow the IMG activity documentation to create a new connection of type G, i.e. HTTP Connections to External Server. You may name it e.g. NW_IDM72. The connection needs to point to the J2EE engine where the Virtual Directory Server web service (the callback service) is deployed as a target system.

Note

Provide target system information, like host name and path prefix for the deployed Virtual Directory Server web service (for example /ids_72/router). Also provide logon parameters, i.e. user/password authentication (basic authentication) using the user and the password configured for the Virtual Directory web service upon web service creation.

6.5.6.3 Maintaining Connectors and Connection Types

Context

You need to define the connector itself within the GRC system. You need to define connection type for the connector and assign the connector to a connector group.

Procedure

1. In the structure of your GRC system, navigate to Common Component Settings/Integration Framework.
2. Start the IMG activity Maintain Connectors and Connection Types.
3. Follow the IMG activity documentation to define the connector with a connection type SPML1. Enter values **NW_IDM72** as Source Connector and **NW-IDM72** as Logical Port.
4. Follow the IMG activity documentation to define the connector group (e.g. NW_IDM_GRP) with connection type SPML1.
5. Assign Logical Group in Assign Connector Groups to Group Types.
6. Assign target connector NW_IDM72, created in the previous section, with connection type SPML1 in Assign Connectors to Connector Groups.

6.5.6.4 Maintaining Connection Settings

Context

You need to assign the connector to an integration scenario.

Procedure

1. In the structure of your GRC system, navigate to Common Component Settings/Integration Framework.
2. Start the IMG activity [Maintain Connection Settings](#).
3. Follow the IMG activity documentation to define the integration scenario/work area PROV and assign the defined connector NW_IDM72 to the scenario.

Note

Select the *PROV* line in the *Subscenario* definition and select *Scenario-Connector Link* in the dialog structure to assign the connector to the scenario.

Note

It may be necessary to assign the connector to integration scenarios/work areas AUTH, ROLMG and SUPMG in addition to PROV.

6.5.6.5 Maintaining Connector Settings in Access Control

Context

You need to create the connector and maintain the connector settings for the system connected to the access control application.

Procedure

1. In the structure of your GRC system, navigate to Governance, Risk and Compliance/Access Control.
2. Start the IMG activity [Maintain Connector Settings](#).
3. Follow the IMG activity documentation to create connector NW_IDM72 with application type 11.

6.5.6.6 Maintaining Mapping for Actions and Connector Groups

Context

You need to assign the actions to the connector group and select the default connector for the group.

Procedure

1. In the structure of your GRC system, navigate to Governance, Risk and Compliance/Access Control.
2. Start the IMG activity Maintain Mapping for Actions and Connector Groups.
3. Follow the IMG activity documentation to activate the created connector group (e.g. NW_IDM_GRP) with application type 11 under *Maintain Connector Group Status*.
4. Under *Assign default connector to connector group* assign action **0004 Provisioning** and the created connector NW_IDM72 as the default target connector to the connector group.
5. Assign group field mapping. See the figure below.

Change View "Assign group field mapping": Overview

Connector Group: NW_IDM_GRP
Connector Action: 4

Assign group field mapping

AC Field Name	System Field Name
EMAIL	mail
FIRSTNAME	givenname
USERID	requestuserid
LASTNAME	sn
ROLE_NAME	mskeyvalue
DESCRIPTION	mskeyvalue
USERID	mskeyvalue
ROLE_NAME	PRIVILEGE

6. Assign group parameter mapping and make sure that the parameter EXIT_FROM_GRC is set to **TRUE**. See the figure below.

Dialog Structure

- Maintain Connector Group Status
- Assign default connector to connector group
 - Assign group field mapping
 - Assign group parameter mapping

Connector Group: NW_IDM_GRP
Connector Action: 4

Assign group parameter mapping

Parameter Name	Parameter Value
SEARCH_CRITERIA	IrchBase
CREATE_USER:OC	MX_PERSON
ROLE	privilege
SCHEMA_ID	default
CREATE_USER:userpassword	createpwd
DELETE_USER:OC	MX_PERSON
UNLOCK_USER:islocked	false
UNLOCK_USER:OC	MX_PERSON
LOCK_USER:islocked	true
LOCK_USER:OC	MX_PERSON
ASSIGN_ROLES:OC	MX_PERSON
CHANGE_USER:OC	MX_PERSON
SEARCH_USERS:OC	MX_PERSON
SEARCH_USERS:FILTER_TYPE	present
SEARCH_USERS:unique_identifier	mskeyvalue
SEARCH_USERS:FILTER_BY	objectclass
SEARCH_USERS:FILTER_BY_VALUE	MX_PERSON
SEARCH_ROLES:OC	MX_PRIVILEGE
SEARCH_ROLES:SPML_ID	operation=listprivileges
SEARCH_ROLES:FILTER_TYPE	present
SEARCH_ROLES:FILTER_BY	objectclass
SEARCH_ROLES:FILTER_BY_VALUE	MX_PRIVILEGE
SEARCH_ROLES:unique_identifier	mskeyvalue
SEARCH_USER_ROLES:OC	MX_PERSON
SEARCH_USER_ROLES:FILTER_BY	objectclass
SEARCH_USER_ROLES:FILTER_BY_VALUE	MX_PERSON
ROLE	Privilege
RESET_PASSWORD:OC	MX_PERSON
RESET_PASSWORD:userpassword	resetpwd
password	userpassword
EXIT_FROM_GRC	TRUE
SEARCH_USER_ROLES:SPML_ID	operation=listusers
SEARCH_USER_ROLES:unique_identifier	mskeyvalue
SEARCH_USERS:SPML_ID=operation	operation=listusers
SEARCH_USER_ROLES:SPML_ID	operation=listusers
SEARCH_USER_ROLES:FILTER_TYPE	present

- Save the entries before exiting the IMG activity.

6.5.7 Grouping of Privileges

Grouping of the privileges is done according to the grouping policy defined by the attribute Grouping rule (MX_PRIV_GROUPING_RULE) on the repository (GRC10, or an ABAP repository). Possible settings for this attribute are shown below:

Split on				
Grouping policy/rule	Backend system	Operation type (add/remove)	Privilege attribute	Description
0	Not Applicable	Not Applicable	Not Applicable	No grouping. Every pending object will trigger the GRC provisioning framework and will be sent in its own request to SAP Access Control.
P:7	Yes	Yes	Yes	Same operations sent to the same system and having the same specific privilege attribute defined are sent in one request.
P:6	Yes	No	Yes	All pending objects sent to the same system with a specific privilege attribute defined are grouped together into one request.
P:5	No	Yes	Yes	Pending objects with same operations and a specific privilege attribute defined are sent in the same request.
P:4	No	No	Yes	All pending objects with a specific privilege attribute defined are grouped and sent in the same request.
P:3	Yes	Yes	No	Different operations will not be sent in the same request. Even the identical operations to different systems will not be sent in the same request. Multiple identical operations to the same system will be grouped and sent in the same request.
P:2	Yes	No	No	All operations to the same system/application will be sent in the same request.
P:1	No	Yes	No	Same operations, regardless of the system/application, will be grouped together and sent in the same request.
P:0	No	No	No	All pending objects, regardless of the operation and the system, are sent in the same request.

When grouping by a specific privilege attribute, this specific attribute has to be defined on the repository (held by the attribute MX_PRIV_GROUPING_ATTRIBUTE). This is relevant for the grouping rules P:4, P:5, P:6 and P:7.

Functional Grouping

Functional grouping applies only to privilege assignments that require compliance check to the target system.

When you assign a number of privileges to a user (separately or through one or more business roles), all privileges that require compliance checks are grouped together for one request to the GRC system. The

request is sent only if all member event processes, approvals or future validities that are defined on the privileges are completed.

For example, if a privilege that requires a compliance check wait for 10 days to be approved, other privileges that require a compliance check will also wait for 10 days, until finally all privileges are grouped for one request to the GRC system.

For more information about the functional grouping and how it differs from the performance grouping, see *Assignment Grouping*.

Related Information

[Assignment Grouping \[page 57\]](#)

6.5.7.1 Configuring the Privilege Grouping on the Repository

Context

To configure the grouping rule P:4 based on the privilege attribute MX_OWNER, do the following in the Identity Management Administration User Interface:

Procedure

1. Log on to SAP Identity Management Administration User Interface at `http://<host>:<port>/idm/admin`.
2. Select the *System Configuration* tab and then *Repositories* from the menu on the left.
3. Select a repository (GRC10, or an ABAP repository).
4. In the *Details of Repository* pane, select the *Constants* tab.
5. Select the repository constant MX_PRIV_GROUPING_RULE.
6. Select options  *Grouping*  *Separate by privilege attribute*  and define the privilege attribute MX_OWNER.

▼ P.4

Details of "MX_PRIV_GROUPING_RULE"

Assignment grouping

☐ No Grouping

☒ Grouping

☐ Separate by operation

☐ Separate by application

☒ Separate by privilege attribute MX_OWNER ▼

▲ Close Details

7. Choose [Save](#).

For more information about privilege grouping, see *Privilege Grouping*.

Related Information

[Assignment Grouping \[page 57\]](#)

6.5.7.2 Consequence of Grouping

The possibility for dynamical setting of all critical parameters of the request is needed when grouping the privileges (pending objects).

The majority of the decisions regarding these critical request parameters may be taken based on the properties of the privilege being provisioned. Each privilege may have different relevant properties, thus making it impossible to send a request with a single valid value.

In the GRC provisioning framework, the properties of the leading privilege (pending object) of the group, i.e. the one that actually triggers the provisioning process, decide the dynamic properties of the request. This may not always fit the customer expectations. In that case, the GRC provisioning framework must be explicitly simplified, i.e. the scripts that are used to achieve the dynamic behavior should be tailored adequately.

6.5.7.3 Grouping across Repositories

In order for the two assignments to belong to the same request group (be grouped together), their corresponding pending value objects must share several properties:

- Mandatory properties
 - User ID of the user that assignment is valid for
 - Audit ID of the job where assignments are performed (assignments executed in the same transaction (with the same Audit ID) can be grouped together, regardless of their repositories). MX_PRIV_USERID

attribute is used as an alternative for the Audit ID, to distinguish jobs executed in the same transaction in cases when AuditID is not available.

- Repository name defined for the privileges that are assigned
- Configurable properties
 - Operation type (grouped by same operation)
 - Application (grouped by same application)
 - Attribute value (grouped by attribute of own choice)

In order to be able to group assignment requests across repositories when the Identity Management is configured for compliancy using the centralized provisioning scenario (which is recommended), two additional properties need to be defined on the task responsible for sending the assignment requests:

- Mandatory: that the task is possible to group ("groupability" of the task).
- Optional: grouping rule that will override the individual grouping rules defined for repositories (leading grouping rule, used if different grouping rules are defined for each repository).

Example

Assume the following, as an example:

- The Identity Management is configured for compliancy using centralized provisioning scenario, where the Identity Management typically contains multiple ABAP repositories each with their own set of privileges.
- Each of these ABAP repositories has defined the same tasks MX_VALIDATE_ADD_TASK and MX_ADD_MEMBER_TASK.
- MX_VALIDATE_ADD_TASK holds a reference to the main task from the GRC provisioning framework, the one that will send requests to GRC checking the compliancy (the process AC Validation).
- A set of privileges is assigned to a user, where the privileges belong to different repositories.
- The privilege assignment requests are expected to be grouped into one single request, across the ABAP repositories.

According to the basic concept of grouping, where only requests belonging to the same repositories are grouped, the grouping across the repositories would fail as we here have requests from several ABAP repositories. Therefore, the attention is turned to the task responsible for sending the assignment requests instead, the task held by the attribute MX_VALIDATE_ADD_TASK which in this case is the process AC Validation.

When this process is the same for all ABAP repositories, as is the case here, then it is possible to group requests from multiple repositories into one single request. The attribute MX_OPERATION_TASKID defined on the pending value object holds the reference to this task. Pending assignments (pending value objects) belonging to different repositories but with the same MX_OPERATION_TASKID value can be grouped together (i.e. even if they don't belong to the same repositories).

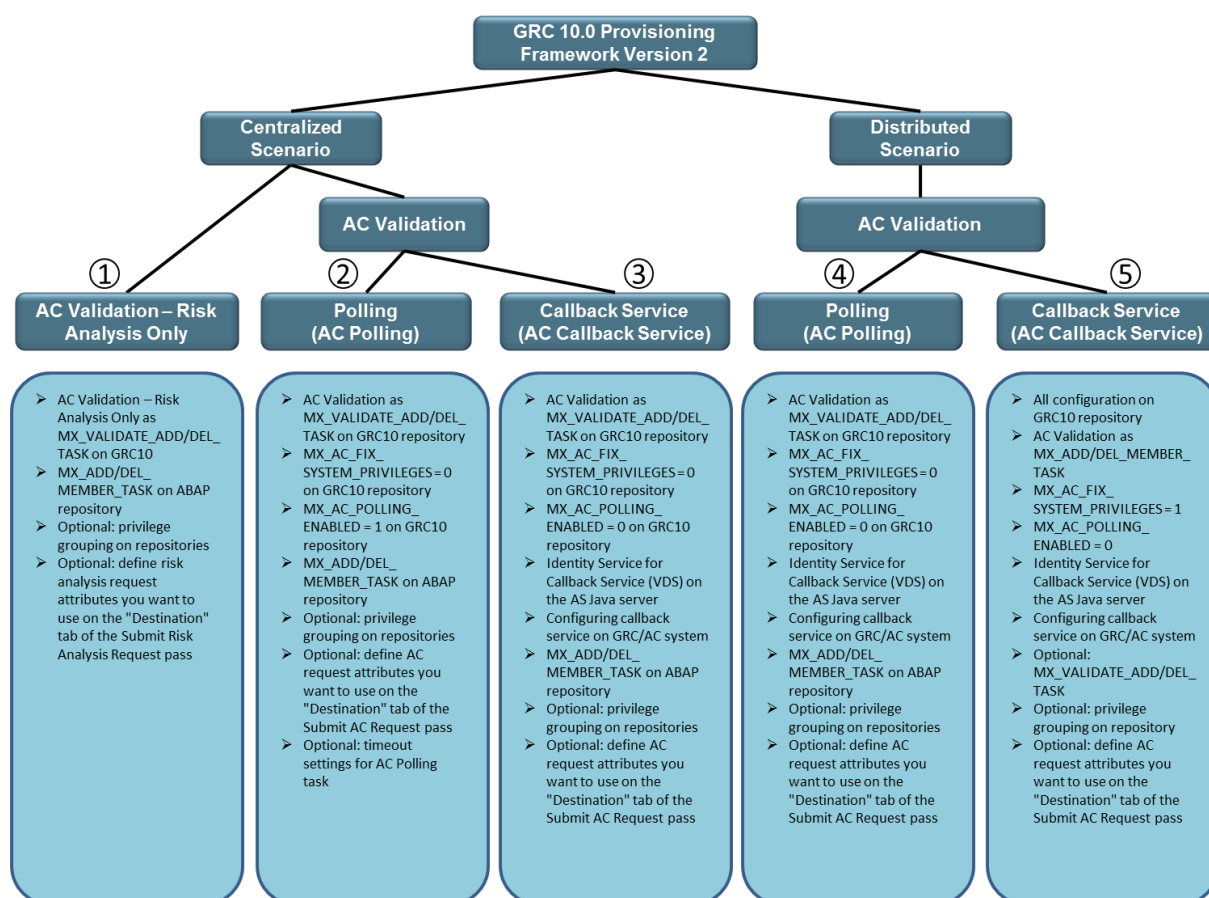
6.5.8 Configuring the Properties on Provisioning Framework Processes

Processes in the GRC 10.0 Provisioning Framework Version 2 may have to be adjusted to fit your solution and your needs. This is relevant for the following processes in the following situations:

- If using the process AC Validation (i.e. distributed provisioning scenario): define which AC request attributes you want to use on the *Destination* tab of the Submit AC Request pass. For more information, see section [Submit AC Request \[page 763\]](#) for details.
- If using the process AC Validation – Risk Analysis Only (i.e. centralized provisioning scenario): define which risk analysis request attributes you want to use on the *Destination* tab of the Submit Risk Analysis Request pass. For more information, see section *Perform Risk Analysis/Submit Risk Analysis Request in AC Validation - Risk Analysis Only [page 772]*.
- If using the process AC Polling (i.e. using polling as a result handling scenario): you may want to define the timeout settings with the retry interval for the process other than the default values. For more information, see section *Timeouts in AC Polling [page 769]*.

6.5.9 Configuration Summary

The following illustration summarizes the configuration for different integration scenarios:



6.6 Running the Initial Load Service Jobs

Context

Before any provisioning or deprovisioning through the provisioning framework can be carried out, the framework must be initialized by running the Initial Load service jobs.

In order to successfully hand over the requests to SAP Access Control, the SAP Identity Management requires information about the environment managed by SAP Access Control. During the initialization phase, the Identity Center obtains information about the environment managed by SAP Access Control (managed systems and available roles). Appropriate Identity Management objects (privileges) are created or enhanced, and stored in the identity store, i.e. the information is stored as privileges in the Identity Center identity store.

This is achieved by running the Initial Load service jobs, which are connecting directly to the target system and retrieving the (role/privilege) information. SAP Identity Management executes among others Search Role web service call towards SAP Access Control. The Search Role web service obtains information about roles available for each of the systems managed by SAP Access Control. This call will typically enrich the role information which is already available in the Identity Management.

There are three Initial Load service jobs – one for each of the two landscape configuration scenarios (centralized and distributed) and one which is common for both. All jobs are independent of what result handling scenario is to be implemented (polling or event based/callback).

To run the jobs, do the following:

Procedure

1. Log on to Identity Management Administration User Interface at `http://<host>:<port>/idm/admin` URL.
2. Select the *System Configuration* tab, and choose *Repositories* from the menu on the left.
3. Select a repository of type GRC10.
4. In the *Details of Repository* pane, select the *Jobs* tab and then select the job you want to run.
5. Choose *Run Now*.

Note

The Initial Load service job AC 10.0 – Initial Load – Commons needs to be modified before it is run when using GRC 10.0 Provisioning Framework Version 2. Disable the pass Create AC System Account in the job and then run the job.

Related Information

[AC 10.0 – Initial Load – Commons \[page 751\]](#)

[AC 10.0 – Initial Load – Centralized Provisioning \[page 753\]](#)

[AC 10.0 – Initial Load – Distributed Provisioning \[page 754\]](#)

[Troubleshooting \[page 755\]](#)

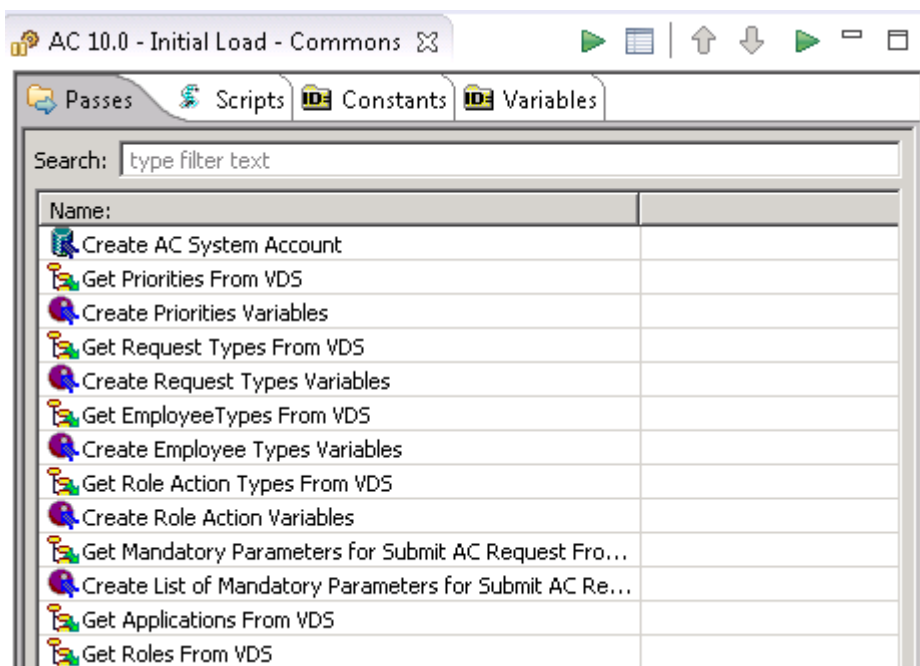
6.6.1 AC 10.0 – Initial Load – Commons

Note

For centralized provisioning scenario, the Initial Load service job AS ABAP – Initial Load job (from the SAP Provisioning Framework) needs to be run before AC 10.0 – Initial Load – Commons.

The job AC 10.0 – Initial Load – Commons needs to be executed disregarding of what provisioning scenario is implemented, centralized or distributed.

It contains the following passes:



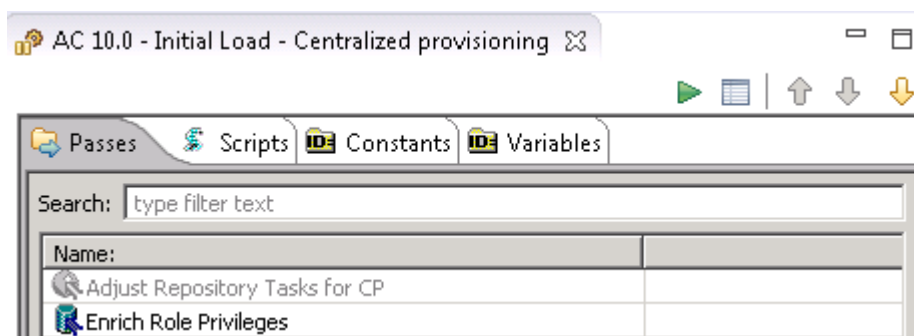
Pass	Description
Create AC System Account	Creates an account that is needed for the approval process triggered by the framework. For this purpose the MSKEY-VALUE held by the attribute MX_AC_APPROVAL_SYSTEM-ACCOUNT on the repository definition GRC10 is used.  Note When running the Initial Load service job AC 10.0 – Initial Load – Commons with GRC 10.0 Provisioning Framework Version 2, disable the pass Create AC System Account in the job before running it. This pass must be disabled before the job is run for the GRC 10.0 Provisioning Framework Version 2.
Get Priorities From VDS	The information about the priorities is obtained from the Virtual Directory Server using the web service call GRAC_LOOKUP_WS. The information is stored in the local table sapGRC10priorities .
Create Priorities Variables	Uses the script sap_processPriorityTypes to process the information stored in the local table sapGRC10priorities and construct the package (priority) variables. A typical example is GRC_ROLE_PRIORITY_HIGH with value 010 .
Get Request Types From VDS	The information about the request types is obtained from the Virtual Directory Server using the web service call GRAC_LOOKUP_WS. The information is stored in the local table sapGRC10requesttypes .
Create Request Types Variables	Uses the script sap_processRequestTypes to process the information stored in the local table sapGRC10requesttypes and construct the package (request) variables. A typical example is GRC_ROLE_REQUEST_NEW_ACCOUNT with value 001 .
Get EmployeeTypes From VDS	The information about the employee types is obtained from the Virtual Directory Server using the web service call GRAC_LOOKUP_WS. The information is stored in the local table sapGRC10employeetypes .
Create Employee Types Variables	Uses the script sap_processEmployeeTypes to process the information stored in the local table sapGRC10employeetypes and construct the package (employee) variables. A typical example is GRC_ROLE_EMPLOYEE_PERMANENT with value 002 .

Pass	Description
Get Role Action Types From VDS	The information about the action types is obtained from the Virtual Directory Server using the web service call GRAC_LOOKUP_WS. The information is stored in the local table <code>sapGRC10roleactiontypes</code> .
Create Role Action Variables	Uses the script <code>sap_processActionCodes</code> to process the information stored in the local table <code>sapGRC10roleactiontypes</code> and construct the package (role action) variables. A typical example is <code>GRC_ROLE_ACTION_ASSIGN</code> with value <code>006</code> .
Get Mandatory Parameters for Submit AC Request From VDS	The mandatory parameters for the Submit AC Request are obtained from the Virtual Directory Server using the web service call <code>GRAC_EUP_CONFIG_DATA_WS</code> . The information is stored in the local table <code>sapGRC10eupconfig</code> .
Create List of Mandatory Parameters for Submit AC Request	Uses the script <code>sap_processEUPConfigResult</code> to process information stored in the local table <code>sapGRC10eupconfig</code> and create the package variables for the mandatory parameters needed for the Submit AC Request task.
Get Applications From VDS	The information about applications managed by SAP Access Control is obtained using the web service call <code>GRAC_SELECT_APPL_WS</code> . The information is stored in the local table <code>sapGRC10applications</code> .
Get Roles From VDS	For each application, the information about roles available in SAP Access Control is obtained using the web service call <code>GRAC_SEARCH_ROLES_WS</code> . The information is stored in the local table <code>sapGRC10roles</code> .

6.6.2 AC 10.0 – Initial Load – Centralized Provisioning

For centralized provisioning scenario, the Initial Load job `AC 10.0 – Initial Load – Centralized provisioning` must be executed (in addition to the `AS ABAP – Initial Load` job (from the SAP Provisioning Framework) and `AC 10.0 – Initial Load – Commons` job executed previously).

The AC 10.0 – Initial Load – Centralized provisioning job contains the following passes:



Pass	Description
Adjust Repository Tasks for CP	The pass is for future use and is disabled by default.
Enrich Role Privileges	The attributes needed by the provisioning framework need to be appended to the corresponding ABAP roles to each of the SAP Access Control roles:

	Attribute	Value
.	MSKEYVALUE	\$FUNCTION.sap_matchACRoleToIdMPrivilege(%application%!!%name%)\$\$
	MX_APPLICATION_ID	%application%
	MX_AC_ROLEID	%name%
	MX_AC_ROLETYPE	%type%
#	--- activate description to be...	
#	DESCRIPTION	%description%
#	--- add task references if AC...	
#	MX_ADD_MEMBER_TASK	
#	MX_DEL_MEMBER_TASK	
	MX_REPOSITORY_VALIDATE	%%rep.\$NAME%

The script `sap_matchACRoleToIdMPrivilege` is used to match the SAP Access Control roles to the SAP Identity Management privileges in the following way:

- Try to find privilege named `PRIV:ROLE:<MX_APPLICATION_ID>:<MX_AC_ROLEID>`.
- If the step above fails, find a repository that holds a constant `MX_AC_APPLICATION_ID` with a value that matches `MX_APPLICATION_ID`. Then try to find privilege named `PRIV:ROLE:<name of the just obtained repository>:<MX_AC_ROLEID>`.

To retrieve the `DESCRIPTION` attribute from the SAP Access Control, the attribute needs to be activated in the pass definitions. Also, in cases where the SAP Access Control only manages a subset of the back-end roles, the attributes `MX_ADD_MEMBER_TASK` and `MX_DEL_MEMBER_TASK` need to be activated and the task references added in the pass definitions.

6.6.3 AC 10.0 – Initial Load – Distributed Provisioning

For distributed provisioning scenario, the Initial Load job `AC 10.0 – Initial Load – Distributed provisioning` must be executed (in addition to the `AC 10.0 – Initial Load – Commons` job executed previously).

The imported `AC 10.0 – Initial Load – Distributed provisioning` job contains the following passes:

Pass	Description
Adjust Repository Tasks for DP	The pass is for future use and is disabled by default.
Create System Privileges for Applications	This pass creates a system application privilege that is added to all accounts provisioned to the application managed by SAP Access Control. These are not real privileges and will not be assignable as all others.
Create Role Privileges	The pass creates MX_PRIVILEGE objects in the identity store, for each of the roles obtained from SAP Access Control. For this, the following information is used: - Information from sap GRC10roles (%application%, %name%, etc). - Name of the repository.

6.6.4 Troubleshooting

If any problems should occur during the execution of the Initial Load jobs (as well as other tasks and jobs in general), you can check some of the following:

- Verify that the dispatcher is running and that it is enabled for the jobs.
- Verify that all tasks and jobs are enabled.
- Verify that the job has been defined for the given dispatcher.
- Verify that the repository is defined on the tasks.
- View the logs.
 - System log
Verify that the dispatcher has requested the given job.
 - Job log
View any error messages in the job log to see if you can find the cause of the problem.
- If you need to investigate a job more thoroughly, you can specify a different log file name for the job in the [Logging](#) tab of the job properties. You can also deselect the check box Reset output file to avoid overwriting the log file each time the job is run. This can be useful when debugging a provisioning job that may be run several times in sequence.
- If you need more logging info from a specific job, you can create a specific dispatcher and increase the log level in the dispatcher's `.prop` file. Specify that the job is to be run by this specific dispatcher. Make sure that the dispatcher is not running. To run the job, start the dispatcher from the command line with the following command: `dispatcher_service_<dispatcher name> test runonce`

The job will then be run once and a detailed log file will be created.

6.7 Timeouts and Stale Requests

There is a configuration option in the SAP Access Control, which allows the user to close any older requests in the system that have been waiting for an approver for a long period of time (stale requests).

This configuration should be aligned with the timeout settings configured in the GRC provisioning framework (for details see section [Timeouts \[page 770\]](#)).

For more information about the configuration of the stale requests, see the documentation for SAP Access Control 10.0.

6.8 Running the GRC Provisioning Framework

You have a fully configured solution for integration between SAP Identity Management 8.0 and SAP Access Control (for SAP Governance, Risk and Compliance (GRC) Solutions) 10.0 using the GRC 10.0 Provisioning Framework Version 2.

To run your solution, you need to assign a privilege to a user from the Identity Management User Interface, using a form for privilege assignment to the user.

For more information about the privilege assignment process, see [Privilege Assignment Process \[page 757\]](#).

6.9 Updating the GRC Provisioning Framework

To update the GRC provisioning framework, see *Updating the Provisioning Framework*.

Note

Because updating the GRC provisioning framework overwrites the existing framework, we do not recommend changing the framework itself, but instead, you should copy the templates to your custom package before you make changes. If you did make changes to the framework, create a copy of the modified package before performing the update. For more information, see: *Copying a Package*.

Related Information

[Updating the Provisioning Framework \[page 491\]](#)

[Copying Packages \[page 204\]](#)

6.10 Privilege Assignment Process

In this section, the privilege assignment/de-assignment process is described for the distributed and the centralized scenario.

Related Information

[Centralized Provisioning Scenario \[page 757\]](#)

[Distributed Provisioning Scenario \[page 758\]](#)

6.10.1 Centralized Provisioning Scenario

This section describes the privilege assignment/deassignment process for centralized scenario.

The existing privileges on ABAP repositories, enriched (by adding properties necessary for operations towards GRC) during the initial load phase, are configured so that any assignment/deassignment to/from an entry is:

- Validated, which means that the privilege is not assigned to/removed from an entry until the request is sent to SAP Access Control and an approval is received.
- Treated as a pending assignment/deassignment: the privilege is not assigned to/removed from an entry until the execution operation is successfully executed.

The following describes the process that takes place when a set of privileges is assigned to/deassigned from a certain user entry:

1. The administrator assigns to/deassigns from a user a set of roles/privileges, or an end-user requests a role/privilege assignment/deassignment.

Note

The privileges associated with the non-SAP repositories (or the SAP repositories that the Identity Center manages itself) normally start a provisioning process that is not a part of this description. Nevertheless, some of the privileges may correspond to the privileges obtained from SAP Access Control in the initial load phase.

2. For the privileges that correspond to the privileges obtained from the GRC/SAP Access Control system, the Identity Management retrieves the value of the process MX_VALIDATE_ADD/DEL_TASK (see the section *Obtaining the Task Values*).
 - If the value for the task is defined, the Identity Management creates a pending object for each of the assigned (removed) privileges.
 - Theoretically, the opposite is possible. But it is not in the scope of this document, since it would lead to a non-compliant provisioning.
3. The Identity Management checks the value of the repository constant MX_PRIV_GROUPING_RULE, of the repository which is defined for validation (GRC10). If necessary, the pending objects are grouped. For more information, see *Grouping of Privileges*.

4. Only one pending object in the group will actually trigger the validation process. The GRC 10.0 Provisioning Framework Version 2 is initially configured to process all pending objects in a group (P:O), and will create a single request to SAP Access Control where all assigned privileges are sent.
5. The validation process (MX_VALIDATE_ADD/DEL_TASK) is started. Depending on the configuration, this will be either the AC Validation – Risk Analysis Only task or the AC Validation process:
 - Using the AC Validation – Risk Analysis Only process, the validation result will be obtained through a synchronous call to GRC/AC system.
 - Using the AC Validation process, the validation result will be obtained through either process AC Polling or process AC Callback Service, i.e. the AC Validation process needs to wait until either AC Polling or AC Callback Service has completed. When AC Polling is used to obtain the validation result, the Identity Management is polling for the result as configured in the process. When AC Callback Service is used, the GRC/AC system is sending the validation result to the Identity Management through a defined Identity Service on the AS Java (VDS), i.e. the Identity Management is waiting until notified. When the AC Polling or the AC Callback Service process has completed, the AC Validation process processes the result and completes.
6. If validation is successful, the Identity Management continues the provisioning procedure, checking and executing the process MX_ADD/DEL_MEMBER_TASK defined for privileges (and their associated repositories).

6.10.2 Distributed Provisioning Scenario

The privileges created in the initial load phase are configured so that any assignment to an entry is treated as a pending assignment, which means that the privilege is not assigned to/removed from an entry until the execution operation is successful (in this case an assignment through SAP Access Control 10.0).

The following describes the process that takes place when a set of privileges is assigned/deassigned to/from a certain user entry:

1. The administrator assigns to/deassigns from a user a set of roles/privileges, or an end-user requests a role/privilege assignment/deassignment.

Note

The privileges that are associated with the non-SAP repositories (or the SAP repositories that the Identity Center manages itself) normally start a provisioning process that is not a part of this description. Nevertheless, some of the privileges may be the privileges obtained from SAP Access Control in the initial load phase.

2. For the privileges that correspond to the privileges obtained from the GRC/SAP Access Control system, the Identity Management obtains the value of add member event process (or remove member event process, depending on the operation). See section *Obtaining the Task Values*. If the value of the MX_ADD/DEL_MEMBER_TASK is defined (process AC Validation), the Identity Management creates a pending object for each of the assigned (removed) privileges.
3. The Identity Management checks the value of the repository constant MX_PRIV_GROUPING_RULE on the repository which the privileges belong to (GRC10), and if necessary groups the pending objects. For more information, see section *Grouping of Privileges*.
4. Only one pending object in the group will actually trigger the provisioning process. The GRC 10.0 Provisioning Framework Version 2 is initially configured to process all pending objects in a group (P:O), and will create a single request to SAP Access Control where all assigned privileges are sent.

5. The MX_ADD/DEL_MEMBER_TASK is started, which is the AC Validation process. The result will be obtained through either process AC Polling or process AC Callback Service, i.e. the AC Validation process waits until either AC Polling or AC Callback Service has completed:
 - When AC Polling is used, the Identity Management is polling for the result as configured in the process.
 - When AC Callback Service is used, the GRC/AC system is sending the result to the Identity Management through a defined Identity Service on the AS Java (VDS), i.e. the Identity Management is waiting until notified.
6. When the AC Polling or the AC Callback Service process completes successfully, the AC Validation process completes successfully. The provisioning has been performed by the SAP Access Control (GRC), and not by the Identity Management, in this case.

6.10.3 Obtaining the Process Values

When performing provisioning/deprovisioning with GRC provisioning framework, the following process and repository reference attributes are involved:

- MX_ADD_MEMBER_TASK
- MX_DEL_MEMBER_TASK
- MX_VALIDATE_ADD_TASK
- MX_VALIDATE_DEL_TASK
- MX_REPOSITORYNAME
- MX_REPOSITORY_VALIDATE
- MX_REPOSITORY_ADD_MEMBER
- MX_REPOSITORY_DEL_MEMBER

MX_ADD/DEL_MEMBER_TASK

For the processes MX_ADD/DEL_MEMBER_TASK the value is obtained in the following way (in the given order):

- From the privilege. If the value defined is **-1**, the further process stops.
- If MX_ADD/DEL_MEMBER_TASK is not defined on the privilege, then from the repository defined by the privilege attribute MX_REPOSITORY_ADD/DEL_MEMBER.
- If MX_REPOSITORY_ADD/DEL_MEMBER is not defined, then from the repository defined by the attribute MX_REPOSITORYNAME.

MX_VALIDATE_ADD/DEL_TASK

For the processes MX_VALIDATE_ADD/DEL_TASK the value is obtained in the following way (in the given order):

- From the privilege. If the value defined is **-1**, the further process stops.

- If not defined on the privilege, then from the repository defined by the attribute `MX_REPOSITORY_VALIDATE`.
- If `MX_REPOSITORY_ADD/DEL_MEMBER` not defined, then from repository defined by the attribute `MX_REPOSITORYNAME`.

6.11 Business Role Assignment Process

In this section, the assignment/de-assignment process of business roles maintained in GRC is described for the centralized scenario with polling.

The existing business roles on ABAP repositories, enriched (by adding properties necessary for operations towards GRC) during the initial load phase, are configured so that any assignment/deassignment to/from an entry is:

- Validated, which means that the business role is not assigned to/removed from an entry until the request is sent to SAP Access Control and an approval is received.
- Treated as a pending assignment/deassignment: the business role is not assigned to/removed from an entry until the execution operation is successfully executed.

The following describes the process that takes place when a business role maintained in GRC is assigned to/deassigned from a certain user entry:

1. The administrator assigns to/deassigns from a user a business role, or an end-user requests a business role assignment/deassignment.
2. For the business roles that correspond to the business roles obtained from the GRC/SAP Access Control system, the Identity Management retrieves the value of the process `MX_VALIDATE_ADD/DEL_TASK` (see the section *Obtaining the Process Values*).
3. The validation process (`MX_VALIDATE_ADD/DEL_TASK`) is started.
Using the AC Validation process, the validation result will be obtained through AC Polling process, that is, the AC Validation process needs to wait until AC Polling has completed. When AC Polling is used to obtain the validation result, the Identity Management is polling for the result as configured in the process. When the AC Polling has completed, the AC Validation process processes the result and completes.
4. If validation is successful, the Identity Management continues the provisioning procedure, checking and executing the process `MX_ADD/DEL_MEMBER_TASK` defined for privileges (and their associated repositories).

6.12 Process Execution Description

This section describes the execution process of the processes in the GRC provisioning framework.

Related Information

[AC Validation \[page 761\]](#)

[AC Polling \[page 769\]](#)

[AC Callback Service \[page 771\]](#)

[AC Validation - Risk Analysis Only \[page 772\]](#)

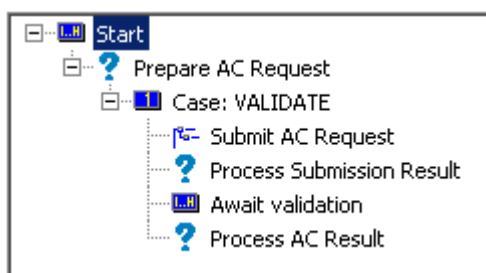
6.12.1 AC Validation

Upon the assignment/deassignment of the SAP Access Control privileges or GRC maintained roles, the Identity Management executes the defined add member event process/remove member event process.

The process AC Validation performs both risk analysis and the provisioning, and its use as such is intended for the distributed scenario. The process can also be used in centralized scenario, where a few adjustments on both the GRC10 repository and on GRC side need to be done in order for the process to be used in centralized scenario. The process AC Validation - Risk Analysis Only is recommended to use in centralized scenario.

The AC Validation process contains the following tasks:

- Prepare AC Request
- Submit AC Request
- Process Submission Result
- Await validation
- Process AC Result



It also configures specific chain result actions.

If the process is successfully executed, as the last step the system will recalculate the necessary system privileges that the entry should have (with the utility task Reassign System Privileges (if needed)).

If any of the tasks in the chain results in error, the utility task Group Failed will be executed. It uses the Java library method `com.sap.idm.grc.ac.ExtACGroupFailed.exec` and removes all pending objects in the particular pending object group without applying them to the user.

6.12.1.1 Prepare AC Request

This task (internal execution task) uses the Java library method `com.sap.idm.grc.ac.ExtACPrepRequest.exec` to:

- Make sure that the links are defined by the MXREF_... attribute, both when adding/removing a privilege/role to/from a user and when adding/removing a user to/from a privilege/role, giving the same pending value object and referenced object in both cases.
- Extract all relevant information from the pending object group.
- Store this information in context variables that are used by tasks in a later processing phase.
- Start the process of constructing an AC request based on the retrieved information.
 - In case a direct privilege is assigned to a user, the AC Validation process continues.
 - In case privileges and/or business roles are assigned to a user through a parent GRC role, the AC Validation process is skipped for the child roles and/or the privileges.
 - In case privileges and/or business roles are not assigned to a user through a parent GRC role, the AC Validation process continues for the child roles and/or the privileges.

The following is a list of context variables that may be of interest:

Variable name	Description
MX_GRC_REQUEST_ID	Request ID (returned to the Identity Management from the SAP Access Control upon the request acceptance).
GRC_REQUEST_ID	As MX_GRC_REQUEST_ID. Used in older versions of the GRC provisioning framework (where the context variable MX_GRC_REQUEST_ID does not exist).
GRCROLEIDLIST	The list of roles approved by SAP Access Control. Format: role-id@system, ...
GRC_RISK_ROLES	A comma separated list of all roles/privileges that contain risks, listed by their names.
GRC_RISK_STATUS	Risk status sent from SAP Access Control (GRC) interpreted by Identity Management. E.g. "OK" or "ERROR".
GRC_RISK_DETAILS	A list of risk details individually for each role/privilege listed in the context variable GRC_RISK_ROLES.
PENDINGMSKEY	The MSKEY of the pending object (the leading in the group).
PENDINGTARGETMSKEY	The MSKEY of the entry that the privilege(s) is /are assigned to.
PENDINGUSERNAME	The MSKEYVALUE of the user entry that the privilege(s) is/are assigned to.
PENDINGPRIVILEGEREPNAME	The repository defined for the privilege.
PENDINGPRIVILEGEMSKEY	The MSKEY of the privilege(s) assigned.
GROUP_PENDINGMSKEYS	Pipe () separated list of all pending value objects (PVOs) in the group, listed by their MSKEYs.

Variable name	Description
GROUP_PENDINGMSKEYS_PRIVILEGEMSKEYS	List of PVO-privilege pairs listed by their MSKEYs (for each PVO listed by the context variable GROUP_PENDINGMSKEYS).
USEDSYSTEMAPPLID	Context variables used by GRC 10.0 Provisioning Framework Version 2.
CLEAN_SYSTEM_PRIVILEGES	
GROUP_SYSTEM_PRIVILEGE_IDS_AND_NAMES	

6.12.1.2 Submit AC Request

This task is responsible for preparing the information that is sent to Virtual Directory Server (that in turn executes a GRAC_USER_ACCESS_WS web service call and sends this information to SAP Access Control).

If the request is accepted, SAP Access Control responds with the request ID that is stored by the Virtual Directory Server in the context variable MX_GRC_REQUEST_ID.

See below how some of the important parameters that are submitted in the request are calculated and filled in.

Common Parameters

DN: Identifier in the console (virtual) tree. The parameter triggers the creation and sending of the web service call GRAC_USER_ACCESS_WS.

changetype: Internal attribute, used for internal processing in the Virtual Directory Server.

Internal Attributes for the VDS

AUDITID: Internal attribute, used for internal processing in the Virtual Directory Server.

Attributes for SAP Access Control

FIRSTNAME, LASTNAME, EMAILADDRESS, TELEPHONE: values sent in the SAP Access Control request. Obtained from the user's entry in the identity store.

Setting the Manager ID

This is only relevant if the web service GRAC_USER_ACCESS_WS requires the manager ID parameter – i.e. if the manager user is responsible for approving the request on the SAP Access Control side. The value NULL is also allowed, in which case the request is forwarded to the default manager.

Whether the manager user is defined as approver or not depends on the configuration of the SAP Access Control system.

The repository of the GRC 10.0 Provisioning Framework Version 2 (GRC10) defines the default manager ID (GRC_MANAGER_ID) that may be used by the task mentioned above (optionally NULL)

The GRC 10.0 Provisioning Framework Version 2 uses the following algorithms to set the values for this parameter:

```
# -- Optional UserInfo Attributes
MGRID $FUNCTION.sap_grc10_getManager($FUNCTION.custom_grc10_getManager()$$$FUNCTION.sap_grc10_getUserManager()$$$rep.GRC_MANAGER_ID%)$$
```

Script sap_grc10_getManager

The script receives three parameters:

1. Output of the script custom_grc10_getManager.
2. Output of the script sap_grc10_getUserManager. The script checks if the user object contains the property MX_MANAGER, and gets the MSKEYVALUE of it if defined.
3. The manager ID configured on the repository GRC10.
The first of these three that returns a value (except an empty String) is used.

Request Properties

All valid requests submitted to SAP Access Control (through web service GRAC_USER_ACCESS_WS) have the following three properties:

- REQUEST_TYPE
- REQUEST_PRIORITY
- REQUEST_EMPLOYEE TYPE

The GRC 10.0 Provisioning Framework uses the following algorithm to set the values for these parameters:

- For each of the parameters, the corresponding script (custom_set<ParameterName>) may be executed. This is an extension point, since the script can return a value based on any customer/proprietary algorithm. This is for example useful if there are various parameters to be sent by the Identity Management and a custom logic is required to decide on the value to choose.
During the initial load phase (running the job AC 10.0 – Initial Load – Commons), the information about valid values for the parameters is obtained from the GRC system, and stored as a set of package variables (see the package variables). The above mentioned custom scripts can, if desired, use these variables.

Setting the Requestor Properties

The web service GRAC_USER_ACCESS_WS requires requestor ID (and optionally the requestor properties) – the identity of the user that performed the role assignment.

The repository of the GRC provisioning framework (GRC10) defines the following default requestor properties that may be used by the task mentioned above:

```
ID= GRC_REQUESTOR_EMAILADDRESS      test.user@sap.com
ID= GRC_REQUESTOR_ID                 TestUser
ID= GRC_REQUESTOR_INITSYSTEM         NW_IDM2
ID= GRC_REQUESTOR_INITSYSTEM2        GP7CLNT520
```

The GRC provisioning framework uses the following algorithm to set the values for the parameters REQUESTORID and REQUESTOREMAILADDRESS:

```
# --- Requestor properties - Request Header Data
REQUESTORID      $FUNCTION.sap_grc10_getRequestorValue(MSKEYVALUE!!%$rep.GRC_REQUESTOR_ID%)$$
REQUEST_INITIATOR %$REP.grc_requestor_initssystem%
REQUESTOREMAILADDRESS $FUNCTION.sap_grc10_getRequestorValue(MX_MAIL_PRIMARY!!%$rep.GRC_REQUESTOR_EMAILADDRESS%)$$
```

Script sap_grc10_getRequestorValue

This script takes two parameters:

1. The attribute name of the parameter whose value is obtained from the user entry that executed the role assignment task.
2. Appropriate default value that is configured on the repository.

The output of this script (hence the parameter value used) is the attribute name if the GRC provisioning framework was able to determine the attribute value in question, or the default value if determining the attribute value was not possible.

The parameter REQUEST_INITIATOR's value (%REP.grc_requestor_initssystem%) is obtained manually in the configuration process.

Setting the Reason

The web service GRAC_USER_ACCESS_WS has an optional reason parameter which will be used in the request sent to SAP Access Control. The value of this parameter is based on the value of the context variable TOTAL_REASON retrieved using the script sap_grc10_getContextVar.

The value of the variable TOTAL_REASON is set in the following way:

- If any of the assigned privileges have an assignment reason configured, these will be retrieved from the pending value object and a single reason string will be created and stored.
- Otherwise, only the default reason for the request is stored. This default reason is configured on the repository GRC10 (the repository constant GRC_REASON).

Mandatory Attributes

Each of the request types mentioned above may have a set of mandatory attributes that must be sent in request.

During the initial load phase (running the job AC 10.0 – Initial Load – Commons), the information about mandatory parameters (and default values, if any) is obtained from the GRC system, and stored as a set of package variables.

In order to be able to send these attributes, check the list of global variables in the Submit AC Request pass of the job with the same name, and enable the corresponding attributes as needed.

The attributes FIRSTNAME, LASTNAME, EMAILADDRESS and TELEPHONE are always sent (even if in theory they do not have to be mandatory).

Obtaining the proper attribute values:

These are recommendations for configuring/obtaining proper attribute values:

- If there is a single attribute value that fits all SAP Access Control requests, configure it as the repository constant (see COMPANY (and MANAGER_ID) in the existing GRC provisioning framework).
- If the value varies with the requestor, obtain the value from user's entry (see DEPARTMENT).
- If the value varies with the role that is provisioned, create a script that retrieves the proper value from the role property.

Considerations for Request Submission

To define the Submit AC Request parameters in the Identity Management correctly, you need to know the corresponding parameters defined on the SAP Access Control side.

Access Request

[Help](#)

Requestor: [TKI](#)

Reason for Request

Request Details

Request Type: *

Request For: *

User: *

Priority:

Business Process: *

Functional Area:

Due Date:

User Access
Risk Violation
Attachments
User Details
Parameters
User Groups

Add Remove Existing Assignments Simulation

Assignment	System	Type	Valid From	Valid To	Owner	Description	Comments	Provisionin...
Z.FI.AR.ER	G17CLNT600	Single Role	26.11.2010	31.12.9999			Add Comm...	Assign

Personal

First Name: *

Last Name: *

Employee Type:

Position:

Academic Title:

Org Data

Manager: [View Details](#)

Job: *

Personnel Number:

Personnel Area:

Organization:

Company

Company:

Function:

Communication

Communication Method:

Telephone:

Email: *

Fax:

Defaults

Start Menu:

Logon Language:

Decimal Notation:

Date Format:

Location

Room Number:

Floor:

Building:

Location:

Logon

Alias:

User Type:

Cost Center:

Accounting Number:

SNC

SNC Name:

Unsecure Logon Allowed(SNC):

Note

Make sure that the mandatory parameters, indicated by an asterisk (e.g. e-mail address), are defined in the Submit AC Request on the Identity Management side.

When a request (with the defined parameters) is received by the SAP Access Control, the request may be approved/rejected through a request administration process.

Access Request

Requestor [TKI](#)

Request submitted successfully. The Request Number is **3162**

[Submit](#) [Close](#)

Access Request :3162 [Help](#)

Request Type [New Account](#) Due Date [01.12.2010](#) Priority [High](#) User [ARNST](#) Requestor [TKI](#) Stage [GRAC MANAGER](#)

[Submit](#) [Print](#) [Other Actions](#) [Close](#) [Refresh](#)

Reason for Request

She needs this role.

Attachments Preview

The table does not contain any data

Request Details

Business Process: * [APO](#)

Functional Area: * [Select](#)

[User Access](#) [Risk Violations](#) [User Details](#) [Parameters](#) [User Groups](#) [Audit Log](#) [External Risk](#) [Comments](#) [Attachments](#)

View: [\[Standard View\]](#) [Simulation](#) [Approve](#) [Reject](#) [Other Actions](#) [Add](#) [Remove](#) [Existing Assignments](#) [Show All Assignments](#) [Filter Settings](#)

Approval S...	Assignment	System	Risk Violati...	Mitigation Co...	Type	Valid From	Valid To	Owner	Comments	Provisioning ...	Provisionin...
Approve	Z FL AR EB	GI7CLNT600	△	△	Single Role	26.11.2010	31.12.9999		Add Comme...	Assign	Not Proces...
Approve											
Reject											

[Submit](#) [Print](#) [Other Actions](#) [Close](#) [Refresh](#)

6.12.1.3 Process Submission Result

This task (internal execution task) uses the Java library method `com.sap.idm.grc.ac.ExtACPostSubmit.exec` to do the following:

- Retrieve the request ID and other related information and store to the pending value object.
- Check attribute `MX_AC_POLLING_ENABLED` of the assigned privilege (defined on the repository of the privilege). If this attribute is set and the polling enabled, then the attribute `MX_AC_POLLING_TASK` is read, and the referenced task executed. Otherwise (if the attribute is not set) the AC Callback Service is enabled and used, and the task stops.

6.12.1.4 Await Validation

This task waits for SAP Access Control to process the request (that is, to approve or reject the privilege or role assignment) . The Await validation task times out in 30 days.

6.12.1.5 Process AC Result

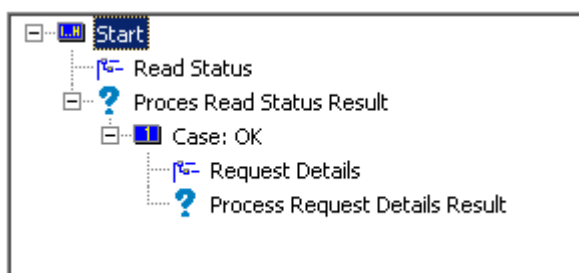
This task (internal execution task) is a part of the process AC Validation and uses the Java library method `com.sap.idm.grc.ac.ExtACProcResult.exec` to do the following:

- If the request is approved, retrieve the result from GRC (either by polling (AC Polling) or a callback service (AC Callback Service) as a result handling scenario) and apply the approved privileges only. Depending on the setting of the repository constant `FIX_SYSTEM_PRIVILEGES`, the appropriate tasks for access control of account existence are either executed (when constant value is **1**) or not (any other values for the constant).
- Make sure that the overall status of the process AC Validation ends either with:
 - OK if everything is ok, i.e. the request is approved, the result obtained (from AC Polling or AC Callback Service) and processed.
 - REJECTED if the request has been declined (a valid action performed by the approver).
 - FAILED (error), if the entire process of obtaining the assignment result fails (e.g. because of a timeout or some other reason (error)). Then the Group Failed task is executed (defined for the AC Validation process as the task to be executed on chain failed result).

6.12.2 AC Polling

The AC Polling process contains one task and one process.

- Read Status
- Process Read Status Result



Read Status

This task polls SAP Access Control for result using the request ID. The polling will continue until SAP Access Control processes the request (i.e. its state read by `getAuditLog` changes to `CLOSED`). The obtained status is stored in the context variable `GRCSTATUS` which will be used in the next phase.

Note

If a request is canceled in GRC, it gets status **Aborted**. The **Aborted** status is supported as of SAP Identity Management 8.0 SP07 Virtual Directory Server patch level 1.

Process Read Status Result

The Process Read Status Result will process the status obtained by the task Read Status. A request can either be approved or declined, or the approval may result in error (e.g. because of the timeout). The overall task status is processed in the following way:

- In case of status OK (request approved), it is possible that only a subset of the roles sent to SAP Access Control is actually approved. In order to obtain information about that, a pass (Request Details) with the call to script `sap_grc10_requestDetails` is executed. It will return a list of privileges that are actually approved. This list is stored in the context variable `GRCROLEIDLIST`.
The internal execution task Process Request Details Result uses the Java library method `com.sap.idm.grc.ac.polling.ExtACProcRequestDetailsResult.exec` and stores the contents of the context variables `GRCSTATUS` and `GRCROLEIDLIST` to a pending value object. After the request is approved, the GRC provisioning framework will apply only the privileges (from the pending object group) that are approved and not the others.
- In case of request being declined by an approver, the content of the context variable `GRCSTATUS` is stored to a pending value object. As no privileges are approved in this case, the list `GRCROLEIDLIST` is empty and thus no privilege information is stored to the pending value object.
- In case of the approval resulting in error (e.g. due to a timeout), the utility task AC Polling Task Failed is executed. This task uses the Java library method `com.sap.idm.grc.ac.polling.ExtACPollingTaskFailed.exec` and sets the failed status on the task.

The behavior is configured on the [Result Handling](#) tab of the AC Polling process – the process AC Polling Task Failed is referenced in field [Execute Process on Failed Result](#).

The screenshot shows the 'AC Polling' configuration window with the 'Result Handling' tab selected. The window has three tabs: 'General', 'Result Handling', and 'Documentation'. The 'Result Handling' tab contains the following configuration fields:

- Execute Process on Initialize:** A dropdown menu showing '-- None --' with a selection button (three dots).
- Chain Result Actions:** A section containing two more configuration fields:
 - Execute Process on OK Result:** A dropdown menu showing '-- None --' with a selection button (three dots).
 - Execute Process on Failed Result:** A dropdown menu showing '4759/AC Polling Task Failed' with a selection button (three dots).

Timeouts

In the Read Status task of the AC Polling process, the timeout settings with the retry interval need to be configured. In case the request is not processed and the status obtained within a defined period of time, the AC

Polling Task Failed process will be executed and set the failed status on the AC Polling process. The goal is to configure sensible timeout and retry interval settings. The following are default values for the Read Status task:

- The required retry interval of the Read Status task is defined to be 1440 retries, every 5 minutes.

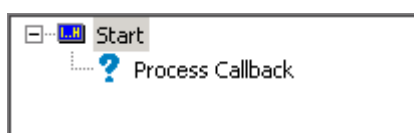
The screenshot shows the configuration for the 'Read Status' task. The 'General' tab is selected. The 'Enabled' checkbox is checked. The 'Task ID/Name' field contains '4766/Read Status'. The 'Job ID' field contains '1889'. The 'Allow Parallel Provisioning' checkbox is unchecked. The 'Scheduling, Dispatchers and Retries' section is expanded, showing 'Schedule Rule' set to 'Provision', 'Run by Dispatchers' with 'idmdispatcher' selected, 'No. of Retries' set to '1440', 'Retry Delay' set to '5' minutes, and 'Right Trim Source Attributes' checked. A red rectangular box highlights the 'No. of Retries' and 'Retry Delay' fields.

In addition, the adequate settings on the SAP Access Control side are needed. See also section *Timeouts and Stale Requests*.

6.12.3 AC Callback Service

The AC Callback Service process contains one task.

- Process Callback



Process Callback

When SAP Access Control has processed the request, it may be configured so it informs the Identity Management about the status of the request. The information about this is sent by executing Exit Web Service (GRAC_EXIT_FROM_IDM_WS) towards the Identity Services.

The Identity Services processes the information in the request and starts the execution of the process AC Callback Service.

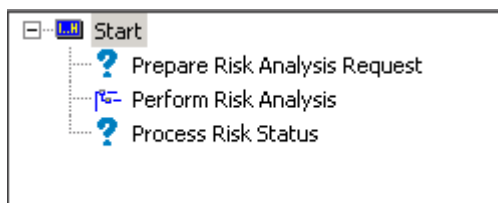
The internal execution task Process Callback uses the Java library method `com.sap.idm.grc.ac.callback.ExtACCallback.exec` to read the information (the request status), process it and save it according to the following:

- In case of request approved, it is possible that only a subset of the roles sent to SAP Access Control is actually approved. A list of privileges that are actually approved is stored in the context variable `GRCROLEIDLIST`. The contents of the context variables `GRCSTATUS` and `GRCROLEIDLIST` are stored to a pending value object. After the request is approved, the GRC provisioning framework will apply only the privileges (from the pending object group) that are approved and not the others.
- In case of request being declined by an approver, the content of the context variable `GRCSTATUS` is stored to a pending value object. As no privileges are approved in this case (as the request is declined), the list `GRCROLEIDLIST` is empty and thus no privilege/role information is stored to the pending value object.

6.12.4 AC Validation - Risk Analysis Only

The AC Validation - Risk Analysis Only process contains three tasks.

- Prepare Risk Analysis Request
- Perform Risk Analysis
- Process Risk Status



This process has some similarities to the AC Validation process. The difference is that the AC Validation process performs both risk analysis and provisioning and AC Validation - Risk Analysis Only performs only risk analysis. Therefore, the intended usage of the process AC Validation - Risk Analysis Only is the centralized provisioning scenario. It is still possible to use the process AC Validation for the centralized scenario, although not recommended.

→ Recommendation

Although using AC Validation - Risk Analysis Only is the recommended (simpler) way to implement the centralized provisioning scenario, note the following:

- As this process executes a synchronous call to the GRC system and obtains immediate result – either approved or declined, it does not wait for the result of other processes, such as risk mitigation, that might be triggered in GRC. In this case, the result of the risk mitigation will not be synchronized with SAP Identity Management.
- If you want to configure your SAP Identity Management to wait for the result of a process triggered in GRC, we recommend that you use the Polling scenario.

Where it was required to change some parameters and flags both on the GRC10 repository and on the GRC side in order for the AC Validation process to be used for centralized scenario, this is not the case for AC

Validation - Risk Analysis Only. The only thing that needs to be done is to configure the process as the member event process (Validate add process) on the repository GRC10.

Prepare Risk Analysis Request

This internal execution task uses the Java library method `com.sap.idm.grc.risk.ExtRARPrepRequest.exec` to:

- Make sure that the links are defined by the `MXREF_...` attribute, both when adding/removing a privilege/role to/from a user and when adding/removing a user to/from a privilege/role, giving the same pending value object and referenced object in both cases.
- Extract all relevant information from the pending object group.
- Store this information in context variables that are used by tasks in a later processing phase.
- Start the process of constructing an AC request based on the retrieved information.

Perform Risk Analysis

The task Perform Risk Analysis is a part of the AC Validation - Risk Analysis Only process. It contains a job with the same name, which again contains a To LDAP directory pass Submit Risk Analysis Request.

Below, some of the important parameters that the pass Submit Risk Analysis Request submits in the request are described:

Parameters	Description
Common Parameters	DN: Identifier in the console (virtual) tree. The parameter triggers the creation and sending of the web service call for Risk Analysis. Changetype: Internal attribute, used for internal processing in the Virtual Directory Server.
Internal Attributes for the VDS	AUDITID: Internal attribute, used for internal processing in the Virtual Directory Server.
Risk Analysis Web Service Parameters	OBJECTID: MSKEYVALUE of the user. OBJECTTYPE: USR (mandatory, i.e. it has to be set to USR).
Role Simulation Parameters	ROLEDATA: Roles to be sent for risk analysis. SIMULATIONOBJTYPE: ROL (mandatory, i.e. it has to be set to ROL). SIMULATIONRISKONLY: X or empty, normally it has to be X.

Parameters	Description
Optional Parameters	ADDLATTRIB: Although this is an optional parameter, it is recommended to define it. The parameter controls the ways in which the risk analysis is performed on the GRC side, i.e. how strictly the analysis is performed. Consult your GRC server personnel for available values, which are depending on the specific GRC version.

Other parameters are optional, and are used as needed. Consult your GRC server personnel for GRC version specific values. All parameters are single value, except ADDLATTRIB, ORGRULE, ORGVAL and REPORTTYPE. For these parameters, the multiple values are separated by comma (,).

Process Risk Status

This internal execution task uses the Java library method `com.sap.idm.grc.risk.ExtRARProcResult.exec` to process the risk status. The status is handled in the following way:

- An error occurred when processing the risk analysis, i.e. something unpredictable has happened (something went wrong), and no status is obtainable: the utility task Group Failed is executed. It uses the Java library method `com.sap.idm.grc.ac.ExtACGroupFailed.exec` and removes all pending objects in the particular pending object group without applying them to the user (i.e. the roles and privileges will not be assigned).
- The risk analysis is successfully performed: the roles/privileges with the risk are added to the pending value object. Roles/privileges will either be applied if no risks are discovered (and thus the subsequent provisioning tasks will be executed), or declined if any risks are discovered (no provisioning will then take place, i.e. the subsequent tasks will not run).

6.13 GRC Related Repository Constants

The GRC related repository constants can be grouped into the following groups of constants:

- Virtual Directory Server related constants
- Request constants
- Runtime constants
- Business Role Synchronization related constants

Virtual Directory Server Constants

These constants define the values needed for successful communication with the Virtual Directory Server. The following constants have hard-coded values and should not be changed. They can be changed however, but the appropriate values in the template have to be changed accordingly:

Constant Name	Description
VDS2GRC_SUFFIX	The top RDN of the virtual tree in the Virtual Directory Server. The default value is "o=grc".
VDS2GRC_BRANCH_LOOKUPSERVICE	The RDNs of the branches in the virtual tree of the Virtual Directory Server (one branch for each of the exposed web services).
VDS2GRC_BRANCH_REQUESTSTATUS	
VDS2GRC_BRANCH_RISKANALYSIS	
VDS2GRC_BRANCH_ROLEDETAILS	
VDS2GRC_BRANCH_SEARCHROLES	
VDS2GRC_BRANCH_SELECTAPPLICATIONS	
SVDS2GRC_BRANCH_USERACCESSREQUEST	
VDS2GRC_BRANCH_PROVISIONINGLOG	
VDS2GRC_BRANCH_REQUESTDETAILS	
VDS2GRC_BRANCH_EXITSERVICE	
VDS2GRC_BRANCH_EUPCONFIG	

The following values must be configured. Use the values specified when configuring the Virtual Directory Server. For details, see [Configuring the Virtual Directory Server \[page 710\]](#)

Constant Name	Description
VDS2GRC_HOST	The host name or IP address of the Virtual Directory Server.
VDS2GRC_PORT	The LDAP port which is used by the Virtual Directory Server.
VDS2GRC_LOGIN	The credentials of the user used to log on to the Virtual Directory Server. The default value is grcuser/grcuser (user/password). These default credentials in the template can be modified in the configuration.
VDS2GRC_PASSWORD	

GRC Request Constants

These constants are a part of the request sent to SAP Access Control (the web service call GRAC_USER_ACCESS_WS):

Constant Name	Description
GRC_MANAGER_ID	The user ID of one of the existing users in the SAP Application Server Java which runs SAP Access Control. This user acts as the default approver for provisioning request (and must therefore have approval access rights in SAP Access Control). The default value is NULL.
GRC_REASON	The default reason for the request. This is only used if the reason is not set by other means.
GRC_COMPANY	The default company string. Make sure that this company exists in your Access Control configuration. Whether or not the GRC_COMPANY attribute is mandatory for a request depends on your Access Control configuration.
GRC_REQUESTOR_ID	Correct values of one of the valid users in the SAP Application Server Java which runs SAP Access Control. These values are used if it is not possible to obtain them from the user entry that executed the role assignment task.
GRC_REQUESTOR_EMAILADDRESS	
GRC_REQUESTOR_INITSYSTEM	

Runtime Constants

These constants set the values that are necessary for the internal processing of the provisioning framework.

Constant Name	Description
MX_PRIV_GROUPING_RULE	P:0 by default. Used for privilege grouping for the operation(s) involving the GRC10 repository.
MX_AC_FIX_SYSTEM_PRIVILEGES	Legal values are: 0: keeping track of the accounts in the target systems is the responsibility of SAP Provisioning Framework. 1: the GRC 10.0 Provisioning Framework Version 2 keeps track of accounts in various target systems. The default value is "1". The value should not be changed unless using the task AC Validation for centralized provisioning scenario, in which case the value of the repository constant needs to be set to "0".
MX_AC_POLLING_ENABLED	If set to value "1", the GRC 10.0 Provisioning Framework Version 2 will invoke the polling process in order to obtain the result of the request after successfully sending a request to SAP Access Control. The task referenced by attribute MX_AC_POLLING_TASK is executed.

Constant Name	Description
MX_REQ_PRIV	<Privilege reference (MSKEYVALUE)> Master privilege
REPOSITORY_TYPE	GRC (do not change)

Business Role Synchronization Constants

These constants set the values that are necessary for SAP Identity Management to be able to perform function calls remotely.

Constant Name	Description
JCO_CLIENT_ASHOST	<application_server_host> Host name or IP address of a specific AS ABAP
JCO_CLIENT_CLIENT	<client> Client of the communication (technical) user
JCO_CLIENT_GROUP	<logon group>, for example, PUBLIC The name of AS ABAP logon group
JCO_CLIENT_GWHOST	<gateway host> AS ABAP Gateway Host To find out the data for your Gateway, choose Administration > System Administration > Monitor > System Monitoring > Gateway Monitor . A list is displayed of all the SAP Gateways connected to your SAP System. Usually, a Gateway is assigned to each SAP application server and database server.
JCO_CLIENT_GWSERV	<gateway service>, for example, "sapgw00" AS ABAP Gateway Service
JCO_CLIENT_LANG	<language ID>, for example, EN Language of the communication (technical) user
JCO_CLIENT_MSHOST	<message_server_host> Host name or IP address of the message server

Constant Name	Description
JCO_CLIENT_PASSWD	<password> Password of the communication (technical) user. For AS ABAP Release 4.6C, use uppercase.
JCO_CLIENT_R3NAME	<SID> AS ABAP System ID
JCO_CLIENT_SNC_LIB	Path to library which provides SNC service
JCO_CLIENT_SNC_MODE	Secure network connection (SNC) mode Possible values: • 0 (off) • 1 (on)
JCO_CLIENT_SNC_MYNAME	<SNC name> SNC name; overrides default SNC partner
JCO_CLIENT_SNC_PARTNERNAME	SNC partner SNC partner, for example, p:CN=R3, O=XYZ-INC, C=EN
JCO_CLIENT_SNC_QOP	<SNC level of security> Possible values: from 1 to 9
JCO_CLIENT_SYSNR	<system number> AS ABAP system number
JCO_CLIENT_USER	<user_ID>

6.14 Limitations/Tuning of the Framework

The following may be adjusted in the framework:

- The GRC provisioning framework is built around the smallest set of attributes that are required in a GRAC_USER_ACCESS_WS web service call. Submitting additional attributes must be configured as custom attributes on the SAP Access Control side and added to the relevant task(s)/pass(es) of the GRC provisioning framework.
See the section *Mandatory Attributes*.
- The GRAC_USER_ACCESS_WS web service call includes the following information: Request type, priority and employee type. The values that SAP Identity Management uses when executing this call has to be aligned with the values that are configured in SAP Access Control.
See the section *Setting the Requestor Properties*.

The following limitations apply to the GRC provisioning framework:

- Obtaining information about managed systems and roles has to be done regularly - there is no automatic process for this.
- A change of the SAP NetWeaver Identity Management's request (so called remediation) is limited to removing the items. It is not possible to add new roles on SAP Access Control side.

7 Configuring Reports

SAP Identity Management provides several options for reporting on identities. Some of them require integration with SAP solutions, such as SAP Business Warehouse (BW), others require custom configuration, such as creating a job and defining queries to extract personal information. The most recent reporting option - standard reports, is introduced in SAP Identity Management 8.0 SP08.

❗ Note

SAP Identity Management does not support SAP BW/4HANA.

For more information, see:

- [Standard Reports \[page 780\]](#)
- [Identity Reporting Using SAP Business Warehouse](#)
- [How to create HTML Reports](#) 
- [Identity Management reporting via SAP Lumira](#) 

❗ Note

SAP Lumira 1.0 end of mainstream maintenance is August 31, 2018.

You can also create reports by defining an SQL query in a job or using a Display form in SAP Identity Management User Interface. For example:

- Create a regular job, add a pass of your choice (for example, To ASCII) and define an SQL Query to extract personal data from the Identity Management database. See: [Creating a Job](#)
- Use any display form in SAP Identity Management User Interface to obtain information about a particular person.

7.1 Standard Reports

You want to know how to use SAP Identity Management standard reports.

Prerequisites

- You are running SAP Identity Management 8.0 SP08 and you have applied the specified patch of the following components:
 - IDENTITYCENTERCORE 8.0 SP08 Patch Level 15 and higher
 - IDENTITYCENTERRT 8.0 SP08 Patch Level 13 and higher

- IDMIC 8.0 SP08 Patch Level 10 and higher
- IDM_CLM_ECLIPSE 8.0 SP08 version 10 and higher
- You have imported the `com.sap.idm.reporting` package.
- You have the `MX_PRIV:REPORT_EXECUTE` and `MX_PRIV:WD:TAB_REPORT` privileges assigned.
- You have enabled the [Clean up old report entries](#) housekeeping procedure to remove report entries that are older than the specified number of days to keep (by default, 30 days).

Overview

Standard reports are introduced in SAP Identity Management 8.0 SP08. They are delivered as predefined forms in the Identity Management User Interface (UI). You can run them for one specific user, role and privilege or for all users.

Report forms are displayed and run on the [Self Services](#) and [Manage](#) tabs. Reports for one entry are run on the [Manage](#) tab, while reports for all entries - on [Self Services](#) tab. Report results are listed on the [View Reports](#) tab.

For every report, a specific script `sap_generate<report_name>Report` is implemented. It defines the SQL select query for a given report, that is, what data will be extracted in the report table. The script calls the `uCreateReport` uFunction. It takes two parameters: the SQL query and the name of the report file. The function generates a zip file containing the report in a csv file.

When you run a report in the Identity Management UI, a new entry is created for the `MX_REPORT` entry type. This entry (report) is added on the [View Reports](#) tab and provides the following information: the entry the report is being executed for (in case the report is run for all entries, this will be the entry that run the report), the status, the report name, the report date, and the result.

Results are displayed as zip links. Every link title contains the report name and date in the following format: `<Report_Name> yyyy-MM-dd 'T' HH:mm:ss`. You can download the zip file and extract the csv file.

Reports are listed by the date when they are run. The most recent ones are at the top of the list. Report data is extracted from the database in real time.

With standard reports, a housekeeping procedure is introduced: [Clean up old report entries](#), which cleans up report entries older than 30 days. By default, it is not enabled.

List of Reports

The following is a list of all standard reports in the Identity Management UI:

Reports executed for all users and privileges

- [Users Attributes](#)
This report shows you all users with their attributes. The following attributes are not returned in this report: assignment attributes, `MX_USER_PICTURE` attribute and multivalue attributes. The only multivalue attribute that is returned is `MX_MANAGER`.

Note

On Microsoft SQL Server, Oracle and IBM DB2, if more than one values for MX_MANAGER exist, all the managers of a user will be listed in one row separated by "|" pipe symbol.

On SAP ASE, if more than one values for MX_MANAGER exist, every manager of a user will be listed in a separate row. Therefore, on SAP ASE, if you customize a report or create your own, we recommend that you do not use more than one multivalue attribute in a report form.

- [Users Assignments](#)

This report shows you all users role, privilege, group, and dynamic group assignments. It contains the user MSKEY and MSKEYVALUE, the assignment MSKEY and MSKEYVALUE and other parameters, as defined in the `sap_generateUsersAssignmentsReport` script.

Every user assignment is displayed in a separate row. This means that if a user is assigned 10 privileges, 5 roles and 5 dynamic groups, the report will contain 20 rows for each assignment of this user.

- [Double Privilege Assignments](#)

This report shows you all users with double privileges assignments, that is, the same privilege assigned to a user through different parent roles, or the same privilege assigned to a user directly or through a role. The report contains the LinkID, the user MSKEY and MSKEYVALUE, the privilege MSKEY and MSKEYVALUE, the parent role MSKEYVALUE and the direct assignment, as defined in the `sap_generateDoublePrivilegesAssignmentsReport` script.

Note

On Microsoft SQL Server, Oracle and IBM DB2, if a privilege has more than one parent role, all parent roles containing double privileges of a user will be listed in one row separated by "|" pipe symbol.

On SAP ASE, if a privilege has more than one parent role, every parent role containing double privileges of a user will be listed in a separate row.

Reports executed for one entry (user, role or privilege)

- [User History Assignments](#)

This report shows you the current and old privilege and role assignments of a given user. It contains the user MSKEYVALUE, the AuditId, assignment MSKEY and MSKEYVALUE and other parameters, as defined in the `sap_generateUserHistoryAssignmentsReport` script.

- [User History Attributes](#)

This report shows you the current and old attribute values of a given user. It contains the user MSKEY, the AttributeId, the current and the old value, the attribute AuditId and other parameters, as defined in the `sap_generateUserHistoryAttributesReport` script.

Every attribute value of the given user is displayed in a separate row.

- [Assigned Users](#) for a given role.

This report shows you all users assigned to a given role. It contains the role MSKEYVALUE and the user MSKEY and MSKEYVALUE, as defined in the `sap_generateRoleAssignedUsersReport` script.

Every user that is assigned to the given role is displayed in a separate row.

- [Assigned Users](#) for a given privilege

This report shows you all users assigned to a given privilege. It contains the privilege MSKEYVALUE and the user MSKEY and MSKEYVALUE, as defined in the `sap_generatePrivilegeAssignedUsersReport` script.

Every user that is assigned to the given privilege is displayed in a separate row.

Related Information

[Reporting Package \[page 485\]](#)

[uCreateReport \[page 1145\]](#)

[Executing Reports \[page 783\]](#)

[Creating or Modifying Reports \[page 784\]](#)

7.1.1 Executing Reports

You want to know how to execute SAP Identity Management standard reports.

Context

Once you fulfill the prerequisites, described in [Standard Reports \[page 780\]](#), you can execute a standard report as follows:

Procedure

1. Open the Identity Management UI and select the report form you want to execute.
 - a. To execute the *Users Attributes*, *Users Assignments* or *Double Privilege Assignment* report form, select the *Self Services* tab.
 - b. To execute the *User History Attributes* or *User History Assignments* report form, select the *Manage* tab, search for a user, and choose ► *Choose Task* ► *Reports* ▾.
 - c. To execute the *Assigned Users* for a given role or *Assigned Users* for a given privilege report forms, select the *Manage* tab, search for the given role or privilege and choose ► *Choose Task* ► *Reports* ▾.
2. Select the report form and choose *Execute Report*.
3. Select the *View Reports* tab and search for your report.

The most recent reports are listed first.

Depending on the system load at the time you run the report or the volume of data, it may take some time before the report is displayed on the *View Reports* tab.

4. Select your report in the *Results* column and download the zip file.
5. Open the csv file and view your report.

Results

If your Identity Management data contains one of the following symbols: “@”, “-”, “+” or “=”, for example: the MSKEYVALUE of a user starts with @John, in the report, those symbols are preceded with single comma (‘). This is to avoid potential vulnerability from injection attacks.

7.1.2 Creating or Modifying Reports

You want to know how to create or modify SAP Identity Management standard reports.

Context

You can create your own reports or modify the existing ones to meet your business needs. To modify existing reports, you can adjust the SQL select query of a given report, so that it extracts the data you need.

📘 Note

Be aware that if you modify an SAP delivered package, updating it with a new version will overwrite your changes. Therefore, we do not recommend changing the SAP delivered package. Instead, import the package that you want to modify under a new name or create a copy of it and make the changes there. You can also use the custom package for your customizations. See: [Updating the Provisioning Framework \[page 491\]](#)

To create your own reports, proceed as follows:

Procedure

1. In the Identity Management Developer Studio, select a package of your choice and create a process.
2. Add an action task.
3. On the [Scripts](#) tab of the action task, create a new script and define your SQL select query.

🔗 Example

In this example, you create a script to generate a report that shows you all users assigned to a given role.

1. Enter a name for the script and choose [OK](#)
For example: **sap_generateRoleAssignedUsersReport** or another name of your choice.
A template for the script is displayed in the script editor view.
2. Add the script code. For example:

📄 Code Syntax

```
// Main function: sap_generateRoleAssignedUsersReport
```

```
function sap_generateRoleAssignedUsersReport(Par) {
    var separator = "!!";
    var inputParameters = Par.split(separator);
    var roleMskey = inputParameters[0];
    var reportName = inputParameters[1];
    var sql = 'select mcOtherEntryMskeyvalue as RoleMskeyvalue,
mcThisMskey as UserMskey, mcThisEntryMskeyvalue as UserMskeyvalue
from mxiv_link_entry e %NOLOCK% left join mxi_objectclasses o on
e.mcThisEntryType = o.ocId where mcLinkState = 0 and o.ocName
= \'MX_PERSON\' and mcOtherMskey = ' + roleMskey + ' order by
mcAddedTime desc';
    return uCreateReport(sql, reportName);
}
```

- `roleMskey` and `reportName` - retrieved from the `Par` parameter.
- `sql` query - selects all users assigned to the given role.

Note

On Microsoft SQL Server, the select queries should include "with (nolock)" or `%NOLOCK%` variable.

- `return uCreateReport(sql, reportName)` - returns a call to `uCreateFunction(sqlQuery, reportName)`. This function builds the report. It takes two parameters: the SQL query and the name of the report file. The function generates a zip file containing the report in a csv file. The result of the report is saved as a value to the `MX_REPORT_RESULT` attribute.

4. Create a [To Identity store](#) pass and configure it as follows:

1. On the [Source](#) tab, select `MX_REPORT` in the [Source Entry Type](#) field.
2. On the [Destination](#) tab, enter the following attributes and attribute values:

Attribute	Values
MSKEYVALUE	%MSKEYVALUE%

Attribute	Values
MX_REPORT_RESULT	This attribute holds the full report result. Provide a value in the following format: \$FUNCTION.<script_name>(%<report_parameters>)%\$\$, where: <script_name> is the name of the script you have created on the <i>Scripts</i> tab of the action task in Step 3. (%<report_parameters>)% contains the following attributes: - MX_REPORT_ENTRY - holds the entry for which the report is executed, in this example it is the mskey of a role. Note If you want the report to be executed for all entries (not for a specific one), do not provide this attribute as a report parameter. For example, you want to generate a report that shows you all users with their attributes. - MSKEYVALUE - holds the name of the report. Instead of using %MSKEYVALUE% as a parameter holding the report name, you can provide a report name of your choice.
MX_REPORT_FORMAT	This attribute defines the format of the report. Enter: ZIP

See: [uCreateReport \[page 1145\]](#)

Role's users [Assigned role's users]

General
Source
Destination
Delta
Documentation

Destination Identity Store

Identity Store: -- Self --

Entry Type: 3/MX_REPORT

Multivalue Delimiter: |

User Info: JobId=%\$ddm.r

Insert Template

Attribute	Value
MSKEYVALUE	%MSKEYVALUE%
MX_REPORT_RESULT	\$FUNCTION.sap_generateRoleAssignedUsersReport(%MX_REPORT_ENTRY!!%MSKEYVALUE%)\$\$
MX_REPORT_FORMAT	ZIP

5. Save your changes.

6. In the Identity Management Developer Studio, create a form of type *Regular*.
7. Select the *Result Handling* tab and select the process you have created in the *Execute Process on Submit* field.
8. Select the *Attributes* tab and choose the entry type for this form.
9. Select the *Report Form* checkbox to specify that this is a form that is used to generate reports.
10. Configure the access control for this form.
11. Save your changes.

Related Information

[Form Properties \[page 1091\]](#)

7.2 Identity Reporting Using SAP Business Warehouse

This section describes how to set up and use SAP Business Warehouse (BW) for reporting on identities managed by SAP Identity Management.

Prerequisites

- SAP Business Warehouse and corresponding SAP NetWeaver Application Server for Java with usage type "BI Java".
 - The release of both systems must be identical.
 - A specific support package might be required for the SAP NetWeaver BI content package (see below).
- SAP NetWeaver BI content:
 - For SAP Identity Management 8.0, you need SAP NetWeaver 7.3 BI Content Add-On 5 or higher.
- Identity data exists in the Identity Center in SAP Identity Management.
- Knowledge of setting up jobs and connectors in the Identity Center and Virtual Directory Server (VDS) components used with SAP Identity Management.
- Knowledge of SAP Business Warehouse, in particular installing and activating SAP-provided BI content, and using Web templates and reports.
- Your BW user is authorized to run transaction STRUST.

Limitations

Note

SAP Identity Management does not support SAP BW/4HANA.

- The following characters cannot be processed by the BI content upload function:
 - Leading explanation point (!)
Leading explanation points (!) in attribute values are prefixed with an underscore (_) by the BW connector before the values are transferred to the BW system.
 - Number sign (#) as a single character
 - Values that contain characters that contain the hexadecimal values HEX00 to HEX1F
 - Special characters not included in the set of allowed characters
This set of characters can be customized. For more information, see [173241](#) .
- Field lengths for InfoObjects are restricted to 60 characters. Attribute values that are longer than 60 characters are truncated. This includes MSKEYVALUES.

Note

As of SAP NetWeaver 7.4 SP02 and higher, you can manually extend InfoObject field length to 250 symbols. For more information, see [2244071](#) .

- The maximum number of identity stores supported is 9999.
- The maximum number of attributes supported is 99999. This is the total number attributes for all identity stores that are included in the integration with BW.
- Time dependencies in BW are taken into account as follows:
 - Planned changes that have not yet occurred are not visible in the BW reports.
 - Time dependencies in BW apply on a daily basis. Therefore, if multiple changes are made to an attribute on the same day, only the last change is visible in the final result.
 - Deletions are effective starting on the following day and therefore not shown on the day of deletion. If the last change on a day is a deletion, it will be reflected in the results on the following day.
 - For multi-value attributes such as role or privilege assignments, this time dependency applies to each value that is added, changed, or removed.

Example

If the same value is changed multiple times on the same day, (for example, added and then deleted) the last change applied will be displayed as set for that day. If the last modification of the attribute value on the day is an Insert record, any deletions on the same day are ignored and the value is displayed as valid until the next (relevant) deletion, if any. Otherwise, if the last modification of the attribute value on the day is a Delete record, the value is displayed as ending on that day and the deletion is displayed as starting on the next day.

- Modification time stamps are always displayed using the local time of the Identity Center database server. There is no conversion to the time zone of the user that is executing the reports in the BW system.
- By default, the BW reporting for SAP Identity Management does not display the milliseconds of the InfoObjects that are of type Date. If you want to add the milliseconds to the BW reporting, see [2030857](#)  for details.

7.2.1 Overview of the System Landscape

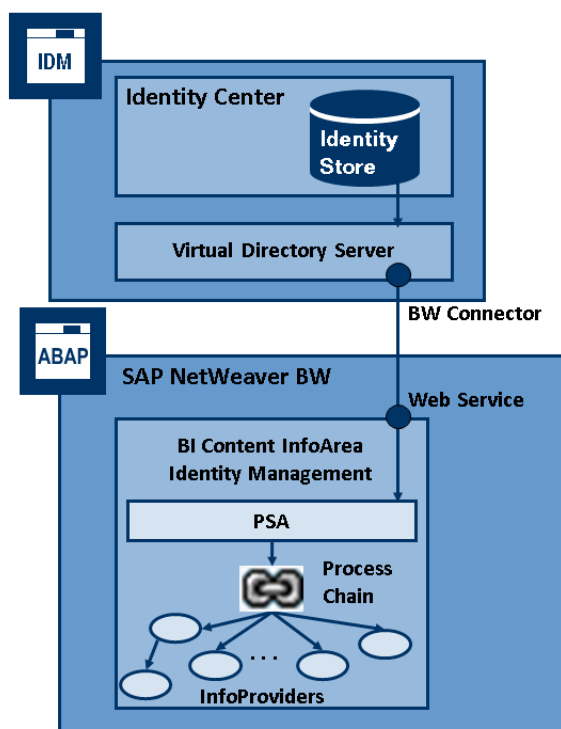
Before you can use BW for reporting on identities, you must set up the connectivity between SAP Identity Management and the BW system. You then have to transfer the data from the identity store in the identity

management system to the BW system. Therefore, before you start with the configuration, you should be familiar with the architecture and system landscape used for the integration.

The following components are needed when using SAP BW for identity reporting:

- Identity Center
- Virtual Directory Server (VDS)
- SAP NetWeaver BW
- Web service on the BW system

See the figure below.



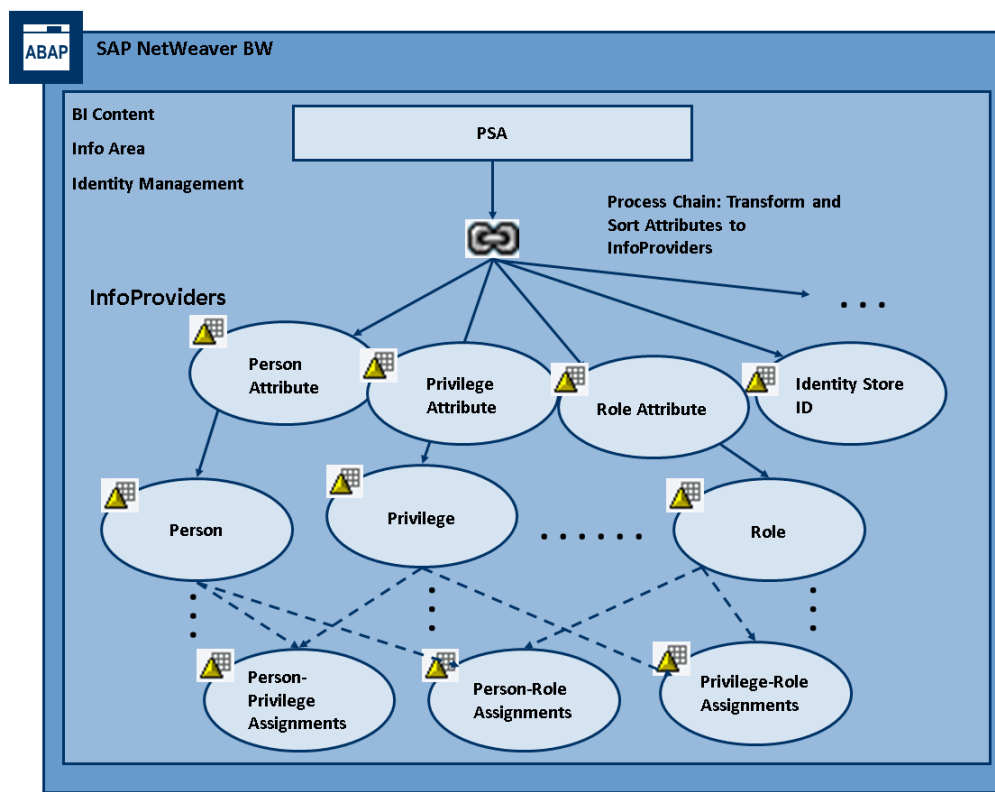
Data Flow

1. The data transfer takes place by starting a job in the Identity Center that issues a Web service call (or multiple calls, depending on the amount of data to transfer) from the VDS to the Persistent Staging Area (PSA) for identity management on the BW system. This applies for both an initial load of the data, as well as for delta loads.
2. Once in the PSA, the identity management process chain is triggered and the identity attributes are transformed and sorted into the corresponding InfoProviders as appropriate.

❁ Example

Attributes that apply to a person such as first name or last name are sorted into the InfoProvider Person. Attributes that apply to a privilege or a role such as the display name are sorted into the InfoProvider Privilege or Role, respectively.

See the figure below.



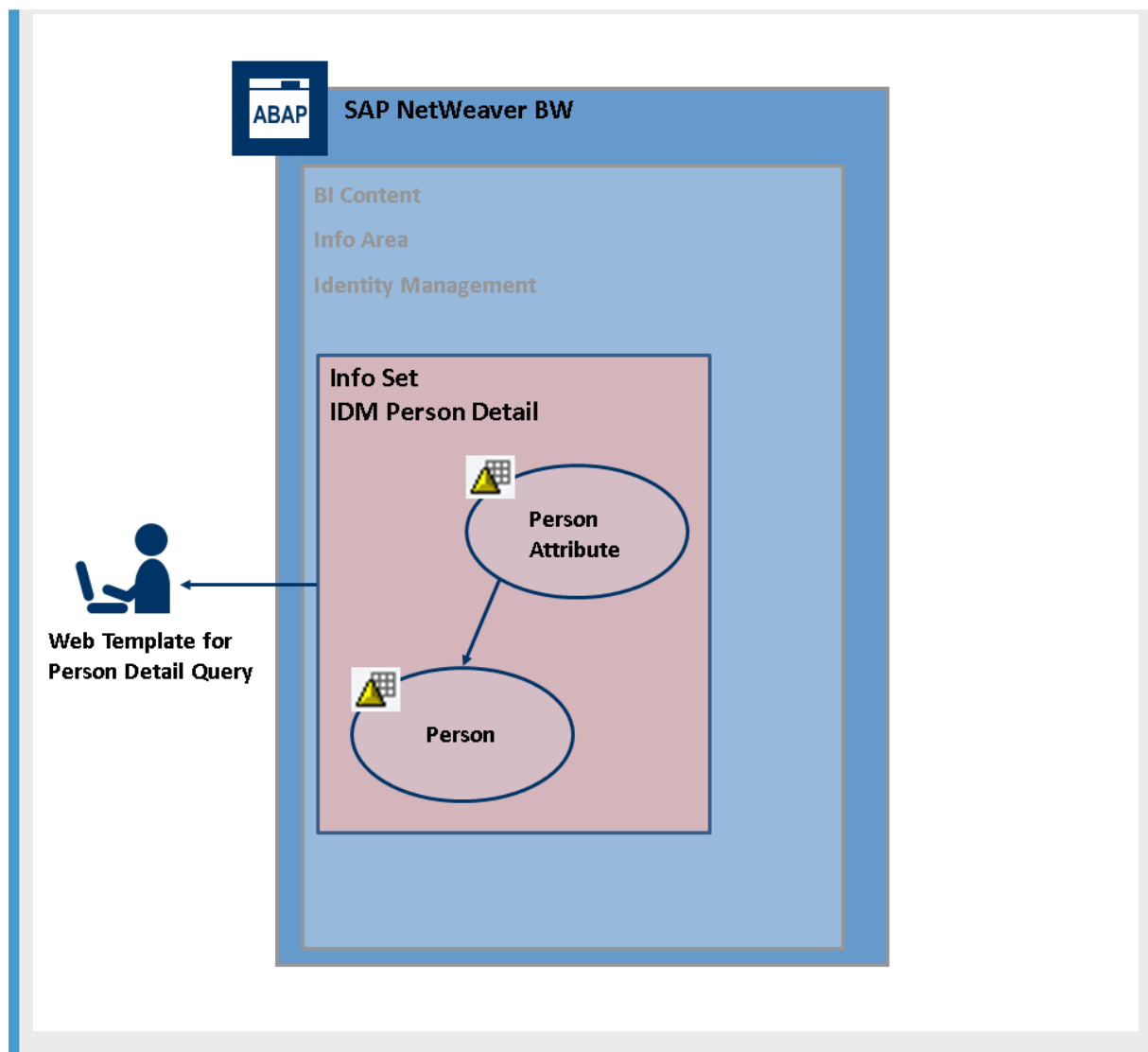
InfoSets and Queries

These InfoProviders are joined into InfoSets that make the appropriate subset of attributes available to the various queries.

❁ Example

The Person Detail query is based on the InfoSet IDM Person Detail, and therefore, has access to the attributes that are assigned to the corresponding InfoProviders Person and Person Attributes.

See the figure below.



Other queries use other sets of attributes and therefore other InfoSets that contain other InfoProviders. InfoProviders may also be included in multiple InfoSets.

The queries can then be called by executing the reports that are accessible from the Web templates. For a list of the SAP-provided Web templates, queries, and the corresponding InfoSets used for identity management reporting, see [Using the Web Templates \[page 808\]](#).

Authorizations for Queries

It is possible to set up the authorizations so that managers and owners of objects are taken into consideration. For these cases, the following roles are available in the BW system:

- **SAP_BW_IDM_REP_ADMIN**
A user with the SAP_BW_IDM_REP_ADMIN role will see all entries requested by the query.
- **SAP_BW_IDM_REP_OWNER**
A user with the SAP_BW_IDM_REP_OWNER role will only see those identity-related objects (persons, roles, privileges, or the corresponding assignments) for which he or she is the owner. The ownership is

determined for the logged-on user for the current point in time based on the identity center attribute MX_OWNER.

- **SAP_BW_IDM_REP_MANAGER**
A user with the SAP_BW_IDM_REP_MANAGER role will only see those identities (and any corresponding assignments that apply to these identities) for which he or she is the manager. The relationship is determined for the logged-on user for the current point in time based on the identity center attribute MX_MANAGER.

For queries that apply to assignments, the user will see all entries where any of the corresponding objects involved in the assignment meets the authorization criteria.

❖ Example

- Example 1: User A owns role X. If user A has the owner authorization for role X, user A will see all of the privileges or persons assigned to role X in the corresponding queries.
- Example 2: User A is the manager for user B and has the manager role. However, user A does not own any roles or privileges. User A will see the role and privilege assignments for user B in the corresponding queries.

More Information

The description and figures above provide an overview about the corresponding architecture and data flow. For more information about the data model used for identity management reporting, see the BI content documentation.

Related Information

[BI Content 7.47](#)

[BI Content 7.57](#)

7.2.2 Overview of the Implementation Steps

The tasks involved with setting up identity reporting using SAP BW are divided into the following categories:

- [Initial Configuration Tasks \[page 793\]](#)
- [Using the Web Templates \[page 794\]](#)
- [Additional Tasks \[page 794\]](#)

7.2.2.1 Initial Configuration Tasks

Context

To initially set up identity reporting using SAP BW, you must do the following:

Procedure

1. Activate the BI content for identity management on the BW system.
2. Configure the Web service on the BW system to automatically start the process chain that processes the data once it is transferred to the BW system.
3. Configure the BW connector on the VDS.
4. Create a repository in the Identity Center.
5. Set up the data transfer job in the Identity Center.
6. Run an initial load for the first transfer of identity data to the BW system.
7. Set up the data transfer job to run periodically.
8. Create the BW analysis authorizations for administrators, owners, and managers, and assign them to roles.

Note

To manually re-send all data to the BW system, repeat an initial load. Note that in this case, to avoid conflicts, you must also restore the initial state in the Identity Center and in the BW system.

7.2.2.2 Using the Web Templates

Once the configuration tasks have been completed, and the data transfer has taken place, you can execute the corresponding queries by using the corresponding Web templates.

7.2.2.3 Additional Tasks

Context

Additional tasks that you may need to perform after initial setup include:

Procedure

1. Checking the progress of the data transfer.
This applies to either initial or delta loads.
2. Starting the process chain manually.
3. Specifying the default identity store to use for queries.
4. Configuring the use of SSL for the connections.
5. Checking the connection parameters to the BW system on the VDS.
6. Customizing the attribute to use for mapping between user IDs in the identity store and the BW system.
7. Release 7.1 only: Adjusting the configuration after updating components.
8. Troubleshooting typical error situations.

7.2.3 Setting up Reporting with SAP BW

Context

To set up identity reporting using SAP BW, you must:

Procedure

1. Activate the BI content for identity management on the BW system.
2. Configure the Web service on the BW system to automatically start the process chain that processes the data once it is transferred to the BW system.
3. Configure the BW connector on the VDS.

4. Create a repository in the Identity Center.
5. Set up the data transfer job in the Identity Center.
6. Run an initial load.
7. Set up the data transfer job to run periodically.
8. Create the BW analysis authorizations for administrators, owners, and managers, and assign them to roles.

7.2.3.1 Activating BI Content for Identity Management

Before you can use SAP BW for identity management reporting, you must copy the BI content for identity management from the delivery version to the active version. This is done by establishing a collection set that contains all of the necessary objects and then installing these objects.

Note

This topic provides the information specific to activating the content for identity management. For more information about the activation concept in general, see the BI content documentation at [Copying BI Content to the Active Version](#) (SAP NetWeaver Release 7.4).

Note

If you have upgraded BI content, delete the master data content from all Identity Management InfoObjects and reactivate the BI content. For more information about deleting the Identity Management content, see [Updating the BI Content](#) section in the SAP Identity Management Migration Guide – Identity Management 7.1 to 7.2.

Prerequisites

Knowledge of BI content and the Data Warehousing Workbench is required.

Procedures

[Specifying the Target Source System \[page 796\]](#)

[Collecting and Activating the BI Content for Identity Management \[page 797\]](#)

7.2.3.1.1 Specifying the Target Source System

Some of the objects to collect require a Web service source system for identity management reporting. Therefore, before you start collecting the objects to install, you must specify this source system.

Context

Proceed as follows:

Procedure

1. Log on to the BW system and start the Data Warehousing Workbench (transaction RSA1).
2. Choose *Modeling* in the left screen area.
3. Select *Source Systems* in the left screen area.
4. From the context menu for *Web service*, choose *Create*.
5. Enter the data as follows:
 - *Logical System Name*: <Logical_System_Name>, for example, **IDM_WS_01**
 - *Source System Name*: <Source_System_Name>, for example, **SAP NetWeaver IDM Reporting**
 - *Type and Release*: **IDM720**

Note

For technical reasons, this value is IDM720, even though it is available as of SAP NetWeaver Identity Management as of Release 7.1 SPS 5. It is still the same for SAP Identity Management 8.0.

6. Save the data.

7.2.3.1.2 Collecting and Activating the BI Content for Identity Management

Once you have specified the source system, you can collect the objects to install.

Context

Proceed as follows:

Procedure

1. In the left screen area of the Data Warehousing Workbench, select *BI Content*.
2. Select the source system assignment to use for selecting the objects to install.
 1. Choose **Edit** > *Source System Assignment* .
A list of available source system assignments appears.
 2. Deselect any pre-marked source systems and select only the source system that you created in the last step (*SAP NetWeaver IDM Reporting, IDM_WS_01*).
 3. Choose *Continue*.
3. The best container to use for collecting the necessary identity management objects is the administrator role.

All of the Web templates and queries are associated with this role, and along with these, the corresponding InfoSets and InfoObjects. Therefore, insert this role into the set of collected objects. Proceed as follows:

 1. In the left screen area, select *Roles*.
 2. In the center screen area, expand *Other Roles* and navigate to the role *Identity Management: Administrator (SAP_BW_IDM_REP_ADMIN)*.
 3. From the context menu for this role, select *Insert Objects for Collection*. You need to choose one or more source systems. In the dialog box, you can select Identity Management reporting, too.
All of the objects that are associated with this role are added to the list of collected objects.
4. The process chain is not included in the set of objects that are associated with the role. Therefore, you must insert it manually.
 1. In the left screen area, select *Object Types*.
 2. In the center screen area, expand *Process Chain* and select *Select Objects* with a double-click.
The list of process chains appears.
 3. Search for *Object Name* *OIDM_DELTA_P001*.
 4. To select the process chain *Identity Management: Delta Load (OIDM_DELTA_P001)* click on the line and choose *Transfer Selections*.
5. Insert the Identity Management Characteristics Catalog into the set of collected objects.
 1. In the left screen area, select *Object Types*.
 2. In the center pane, expand *InfoObject Catalog* and select *Select Objects* with a double-click.
 3. In the same way as for the process chain, search for *Object Name* *OIDM_CHA01*. To select this item, click on the line and choose *Transfer Selections*.

6. Insert some standard InfoObjects for authorization checks.

Note

These InfoObjects may already be installed if you have activated BI content for other scenarios previously.

1. In the left screen area, select *Object Types*.
2. In the center pane, expand *InfoObject Catalog* and select *Select Objects* with a double-click.
3. In the same way as for the process chain, search for the following InfoObjects: *OTCAACTVT*, *OTCAIPROV*, *OTCAVALID*, *OTCAKYFNM*. To select each one, click on each line and choose *Transfer Selections*.
7. Finally insert the BEx query variables required for authorization checks.
 1. In the left screen area, select *Object Types*.
 2. In the center pane, expand **► Query Elements ► Variable ►** and select *Select Objects* with a double-click.
 3. In the same way as for the process chain, search for the following Query Variables: *OS_IDM_GET MANAGERS*, *OS_IDM_GET OWNERS*. Select each one and choose *Transfer Selections*.
8. To make sure that also all dependent objects are included when activating the content, choose *Grouping* at the top of the right screen area and then select *In Dataflow Before and Afterwards*.
9. To activate the content, choose  (Install) and select the appropriate installation option.

Note

The installation process may take some time, therefore, consider running it in the background.

Once completed, the objects needed for identity management are installed and activated in the BI content area for identity management.

7.2.3.2 Configuring the Web Service to Trigger the Process Chain

Once the identity data has been transferred to the BW system from SAP Identity Management, the process chain needs to be triggered that sorts the attributes, performs any necessary transformations, and assigns the attributes to the corresponding InfoProviders. You can either trigger this process manually, or you can use a Web service that does it automatically. The Web service to use is delivered with the BI content add-on, but you must activate it.

Prerequisites

You have the authorization to use the SOA Manager. The corresponding ABAP role is *SAP_BC_WEBSERVICE_ADMIN_BIZ*.

Procedure

1. In the BW system, start the SOA Manager (transaction SOAMANAGER). The SOA Management application starts in a Web browser.

The SOA Management application starts in a Web browser.

2. Navigate to the service administration. As of SAP BW Release 7.3: ► [Service Administration](#) ► [Web Service Configuration](#) ►.

The screen element names vary according to release. Some interfaces also display the terms in brackets.

3. Search for the [Service \(Definition\)](#) **RS_BCT_IDM_CHAIN_START**.
4. Select this service from the search results and choose [Apply Selection](#).

The details for the Web service **RS_BCT_IDM_CHAIN_START** are displayed.

5. Select the [Configurations](#) tab.
6. Choose [Create Service](#).
7. In the screen that appears, enter the data as follows:

- [Service Name](#): **RS_BCT_IDM_CHAIN_START**
- [Description Text](#): **<Description>**
- [New Binding Name](#): **default**

8. Choose [Apply Settings](#). The configuration for the Web service definition is displayed.

9. Choose the [Provider Security](#) step.

10. Select the authentication settings:

- For BW 7.3 (and higher):
Under [Transport Channel Authentication](#), select [User ID/Password](#).

11. To create an SSL endpoint:

- For BW 7.3: The term “endpoint” has been replaced by the term “binding”.
 1. Go back to [Configurations](#) and choose [Create Service](#).
As long as you do not enter a new service name, this creates a new binding for the existing service you selected.
 2. Choose [Next](#).
 3. Under [Transport Guarantee](#), choose [HTTPS](#).
 4. Under [Transport Level Security](#), choose [SSL \(https\)](#).
 5. Under [Transport Channel Authentication](#), select both [User ID/Password](#) and [X.509 SSL Client Certificate](#), as appropriate.

12. Save your configurations.

Note

There is a simplified service configuration (available as of SAP BW Release 7.0 EhP 2) to configure the Web service, a different URL for the endpoint is generated. In this case, you must manually set the VDS constant `MX_CHAIN_START_SERVICE_ENDPOINT_FALLBACK` to the correct endpoint as described in [Checking the Connection Parameters to the BW System on the VDS \[page 821\]](#).

7.2.3.3 Configuring the BW Connector on the VDS

The BW connector runs on the VDS and handles the Web service calls that transfer the identity data to the BW system. To set up this connector, see the procedure below.

Prerequisites

- You know the parameters to use to connect to the AS ABAP server for the BW system. These are:
 - Host name of the AS ABAP server
 - SAP system client
 - Whether communication security is to be used to secure the connection (HTTPS/SSL)
 - HTTP/HTTPS Port on the AS ABAP server
 - Authentication information (user ID and password or SSL client authentication)
- The user to use for the connection must exist in the BW system. This user must have the profile `S_BI-WHM_RFC` and the role `SAP_BC_WEBSERVICE_CONSUMER` assigned to it.

Context

To configure the BW connector on the VDS, proceed as follows:

Procedure

1. Start the Virtual Directory Server.
2. To create a new configuration for the BW connector, choose **File > New**.
3. Under *Group*, select *BW Connector*. Then, under *Template*, select *BW Connector.xml*.
4. Fill in the template with the connection parameters to the BW system.

If communication security is to be used for the connection and you know the SSL-relevant information to use (for example, the location of the truststore and keystore files to use), you can also enter this information in the connection parameters at this time.

5. Choose *OK* and save the configuration to an XML file.
6. If you need to configure the number of entries the BW connector sends to the BW system or the date/time format to be sent to the BW system, open the XML file and select *Constants*. Configure the following:

Constant	Value	Description
SUBMIT_SIZE	Default: 1000	By default, the BW connector sends 1000 entries to the BW system. Depending on your needs, you can reduce the number of the entries. For example, 200 .
IN_DATE_FORMAT	Enter a value in the following format: yyyy-MM-dd HH:mm:ss	Specifies the date/time format to be sent to the BW system. For example, 2015-09-15 12:34:16 .
OUT_DATE_FORMAT	Enter a value in the following format: yyyyMMddHHmmss	Removes special symbols and converts the date/time format into a string of numbers. For example, 20150915123416

7. Create the user that will be used for the connection from the Identity Center to the VDS.
 - a. Expand ► [Virtual Directory Server](#) ► [User groups](#) ►.
 - b. From the context menu for the *Authenticated* node, choose [New....](#)
 - c. Enter a [Login name](#): and choose [Reset...](#) to specify a password.
 - d. Enter a password, confirm, and choose [OK](#) to continue.
 - e. Choose [Apply](#).
8. Start the BW Connector on the VDS.
9. If you are using SSL for the connection, continue with [Configuring SSL for the Web Service Call from the VDS to the BW System \[page 814\]](#).

7.2.3.4 Setting Up the Data Transfer Job on the Identity Center

The next step is to import and set up an Identity Center job to transfer the identity data to the BW system. To do this, you need to import the BW connector package `com.sap.idm.connector.bw` that contains the [IDM to BW Data Transfer](#) job.

Prerequisites

- The Web service that automatically triggers the process chain execution has been activated on the BW system.

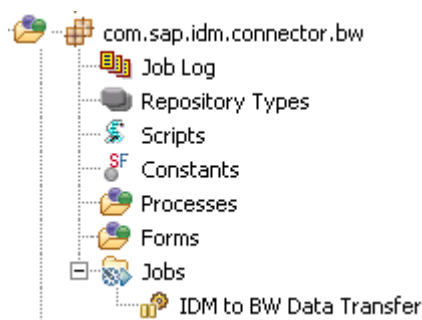
❗ Note

If this Web service is not activated, you can avoid errors by modifying the *End of Transmission* job pass destination that contains the flag that triggers the process chain. Set the parameter `StartChain=<blank>` or disable it by selecting the number sign (#) in the first column for this parameter.

Procedure

1. In the Identity Management Developer Studio tree view, select the identity store where you want to import the BW connector package.
2. Select the *Packages* folder in the tree view and choose *Import* from the context menu.
3. Navigate to the location of the `com.sap.idm.connector.bw.idmpck` package file (for example, `<Install_folder>\Core\ConfigurationPackages\SAP Provisioning Framework`), select it and choose *Open*.
4. In the *Enter Import Reason* dialog box, select *Import* and choose *OK*.

The structure of the imported BW connector package looks like this:



5. Select *Constants* and choose *Properties* from the context menu. Enter the constants value:

Constant	Description
VDS2BW_HOST	Hostname for VDS
VDS2BW_LOGIN	Logon ID for VDS
VDS2BW_PASSWORD	Logon password for VDS
VDS2BW_PORT	LDAP port for VDS
VDS2BW_SUFFIX	Virtual LDAP base DN for BW reporting data

6. Select the job template *IDM to BW Data Transfer* and choose *Properties* from the context menu. On the *General* tab, make sure the job is enabled.
7. Choose **File** > *Save* to save the properties.

Results

The job is set up in the Identity Center. By default, the connection from the Identity Center to the VDS is not secured. If you want to use SSL for this connection, see [Configuring SSL for the Connection from the Identity Center to the VDS \[page 818\]](#).

The next step is [Running an Initial Load \[page 803\]](#).

7.2.3.5 Running an Initial Load

The first time the data transfer job runs, all of the identity data is transferred to the BW system. If, for some reason, you must repeat an initial load, then you must first restore the initial state and then run the job transfer job again. Otherwise, only delta information is transferred. See the procedures below.

Prerequisites

- The *IDM to BW Data Transfer* job has been set up in Identity Center.
- The BW connector is configured on the VDS.
- The VDS is running.

Restoring the Initial State

Context

You only have to perform this part of the procedure if you are repeating an initial load.

Procedure

1. In the Identity Center, delete all of the job variables for the *IDM to BW Data Transfer* job.
2. Delete the identity data in the BW system.
 - a. Disable any previous PSA requests.
 1. In the Data Warehousing Workbench (transaction **RSA1**), navigate to the *Identity Management Entries* DataSource and choose *Manage* from the context menu.
 2. Set the status for each request to *Status NOT OK*.
 - b. Delete data from the central identity management DataStore object.
 1. Navigate to the *IDM Entry Attribute* DataStore object.
 2. From the context menu, choose *Delete Data*.
 - c. Delete the master data for all of the identity management related InfoObjects.

⚠ Caution

The utility program does not ask for confirmation, therefore, make sure you really want to delete the master data from all InfoObjects before performing the following steps.

1. Call transaction **SA38** and enter the program **RSBCT_IDM_DEL_IOBJ_MASTER_DATA**.
2. Run the program and wait until it has finished.

Running the Initial Load Job

Procedure

1. In the Identity Center, navigate to and select the data transfer job *IDM to BW Data Transfer*.
2. Make sure the job is enabled and a dispatcher is activated.
3. Choose Run now to run the job.

Results

The job transfers the identity data to the BW system. To check the progress, see [Checking the Progress of the Data Transfer \[page 811\]](#) .

For delta transfer jobs, we recommend scheduling the job to run periodically. See [Configuring the Data Transfer Job to Run Periodically \[page 804\]](#).

7.2.3.6 Configuring the Data Transfer Job to Run Periodically

Once you have executed the initial load, we recommend you set up the data transfer job to run periodically. In this way, identity data is automatically transferred to the BW system on a regular basis.

Prerequisites

- The *IDM to BW Data Transfer* job has been set up in the Identity Center.
- The initial data has already been transferred to the BW system using the initial load.

→ Recommendation

To avoid timing conflicts, we recommend you wait until the initial load has finished processing before you let delta loads run.

Procedure

1. In the Identity Center, navigate to and select the delta transfer job *IDM to BW Data Transfer*.
2. In the *Schedule rule*: dropdown box, select the scheduling option to use. Choose *Edit...* to view the details about the option settings or to specify them in more detail.

→ Recommendation

Initially, we recommend you set up the job to run once a day, preferably in a time frame where load on the system is low, for example, every night at midnight. You can reduce the interval to let the job run several times a day, however, before doing so, make sure the time needed for the Identity Center job and the process chain processing in the BW system allow for it. Otherwise, timing conflicts may occur.

3. Choose [Apply](#) to confirm the settings.

7.2.3.7 Creating BW Analysis Authorizations and Roles for Using the Queries

Use this topic to set up the BW analysis authorizations to use for identity management reporting. Create three BW analysis authorizations that correspond to the roles administrator, owner, and manager. A user with the administrator analysis authorization can view all of the identity management BI content in the corresponding reports. A user with the owner analysis authorization can view information about those people, roles, privileges, or assignments where he or she is the owner or manager.

Procedure

- [Preparing Standard InfoObjects for Authorization Checks \[page 805\]](#)
- [Creating the Administrator Analysis Authorization \[page 806\]](#)
- [Creating the Owner Analysis Authorization \[page 807\]](#)
- [Creating the Manager Analysis Authorization \[page 807\]](#)
- [Assigning the Analysis Authorizations to Roles \[page 808\]](#)

7.2.3.7.1 Preparing Standard InfoObjects for Authorization Checks

Context

These adjustments might already have been applied to your system if you activated BI content for other scenarios previously. See SAP Note [820183](#)  for details.

Procedure

1. On the BW system, call transaction RSD1.
2. Prepare the following InfoObjects for use with authorization checks: 0TCAACTVT, 0TCAIPROV, 0TCAVALID, 0TCAKYFNM.
 1. Next to *InfoObject*, enter the InfoObject name and select *Maintain*.
 2. Confirm warning message Object IOBJ <InfoObject Name> belongs to technical Content; changes will be overwritten.
 3. Switch to the *Business Explorer* tab.
 4. Enable the *AuthorizationRelevant* flag.
 5. Select *Activate* in the toolbar.

7.2.3.7.2 Creating the Administrator Analysis Authorization

Procedure

1. On the BW system, call transaction RSEADMIN.
2. In the *Authorizations* tab, choose according to release, the *Maintenance* or *Ind. Maint.* button.
3. Create the analysis authorization to use for administrators:
 1. Enter a name in the *Authorization* field, for example, **ZIDM_ALL** and choose *Create*.
 2. Enter short, medium, and long texts, for example, **IDM View All**.
 3. Insert five rows with the following characteristics:
 - 0IDM_OWNR
 - 0TCAACTVT
 - 0TCAIPROV
 - 0TCAKYFNM
 - 0TCAVALID
4. Select the following rows and choose  (Full Authorization).
 - 0IDM_OWNR
 - 0TCAIPROV
 - 0TCAKYFNM
 - 0TCAVALID
5. Deselect all rows and select 0TCAACTVT with a double-click.
6. In the screen that follows, in the *Value Authorizations* tab, enter a row with the column values as follows:
 - *Including/Excluding*: **I**
 - *Operator*: **EQ**
 - *Technical Character. (from)*: **03**
 - *Technical Character. (to)*: **<blank>**

7. Save the data.

7.2.3.7.3 Creating the Owner Analysis Authorization

Procedure

1. Return to the main screen for *Maintaining Authorizations*.
2. Enter the name of the administration analysis authorization you created above in the *Authorization* field and choose *Copy*.
3. Enter a name for the owner analysis authorization in the *Copy To:* field, for example, **ZIDM_OWNER**, and choose again *Copy* or *Continue* according to release.
4. Enter short, medium, and long texts, for example, **IDM View Owner**.
5. Select the row for **ØIDM_OWNR** and choose *Details*.
6. In the screen that follows, in the *Value Authorizations* tab, replace the existing row with the following values:
 - *Including/Excluding:* **I**
 - *Operator:* **EQ**
 - *Technical Character. (from):* **\$ØS_IDM_GET_OWNERS**
 - *Technical Character. (to):* **<b!ank>**
7. Save the data.

7.2.3.7.4 Creating the Manager Analysis Authorization

Context

Repeat the procedure above for the manager analysis authorization.

Procedure

1. Copy either the administrator or the owner analysis authorization to one to use for the manager, for example, **ZIDM_MANAGER**.
2. Change the short, medium, and long texts as appropriate.
3. Select the row for **ØIDM_OWNR** and choose *Details*.
4. In the screen that follows, in the *Value Authorizations* tab, replace the existing row with the following values:
 - *Including/Excluding:* **I**
 - *Operator:* **EQ**
 - *Technical Character. (from):* **\$ØS_IDM_GET MANAGERS**
 - *Technical Character. (to):* **<b!ank>**

5. Save the data.

7.2.3.7.5 Assigning the Analysis Authorizations to Roles

Procedure

1. Call transaction PFCG.
2. Copy the roles SAP_BW_IDM_REP_ADMIN, SAP_BW_IDM_REP_OWNER, and SAP_BW_IDM_REP_MANAGER to roles in your own namespace.
3. For each of these roles, do the following:
 1. Edit the role.
 2. Select the [Authorizations](#) tab and choose [Change Authorization Data](#).
 3. Expand the nodes ► [Business Information Warehouse](#) ► [BI Analysis Authorizations in Role](#) ► [BI Analysis Authorizations in Role](#) ► [BI Analysis Authorizations](#) ►.
 4. Edit this node by selecting it with a double-click.
 5. Enter the name of the analysis authorization that applies to the role (ZIDM_ALL for administrator, ZIDM_OWNER for owner, or ZIDM_MANAGER for manager) in the first 'From' value.
 6. Save and close the selection dialog.
 7. Save again.

7.2.4 Using the Web Templates

Prerequisites

Authorizations are set up accordingly in the BW system. Use the following roles in the BW system to control what is viewed by the user that executes the reports:

- SAP_BW_IDM_REP_MANAGER: This role allows a manager to view those identities and the corresponding assignments for which he or she is currently the manager.
- SAP_BW_IDM_REP_OWNER: This role allows the owner of identities, roles, or privileges to view those objects and the corresponding assignments for which he or she is currently the owner.
- SAP_BW_IDM_REP_ADMIN: This role allows the user to view all identities, privileges, roles, and the corresponding assignments.

For more information, see [Overview of the System Landscape \[page 788\]](#) in the section *Authorization for Queries* and [Creating BW Analysis Authorizations and Roles for Using the Queries \[page 805\]](#).

Context

Once the configuration is completed, you can execute the reports from the Web templates to view the identity data. The table below shows the Web templates we provide for identity reporting.

Web Templates, InfoSets and InfoProviders

Web Template	Technical Name	InfoSet	InfoProviders
Systems with Their Privileges and Assigned Persons (State)	0TPLA_0IDM_IS01_Q0001_V01	IDM Person-Privilege Assignment	Person Privilege
Persons with Their System Access and Privileges (State)	0TPLA_0IDM_IS01_Q0002_V01		Person-Privilege Assignment
Persons with Their System Access and Privileges (History)	0TPLA_0IDM_IS01_Q0003_V01		
Systems with Their Privileges and Assigned Persons (History)	0TPLA_0IDM_IS01_Q0004_V01		
Person Details (State)	0TPLA_0IDM_IS02_Q0001_V01	IDM Person Detail	Person Person Attribute
Person Details (History)	0TPLA_0IDM_IS02_Q0002_V01		
Privilege Details (State)	0TPLA_0IDM_IS03_Q0001_V01	IDM Privilege Detail	Privilege Privilege Attribute
Privilege Details (History)	0TPLA_0IDM_IS03_Q0002_V01		
Role Details (State)	0TPLA_0IDM_IS04_Q0001_V01	IDM Role Detail	Role Role Attribute
Role Details (History)	0TPLA_0IDM_IS04_Q0002_V01		
Roles with Their Users (State)	0TPLA_0IDM_IS05_Q0001_V01	IDM Person-Role Assignment	Person Role
Persons with Their Roles (State)	0TPLA_0IDM_IS05_Q0002_V01		Person-Role Assignment
Persons with Their Roles (History)	0TPLA_0IDM_IS05_Q0003_V01		
Roles with Their Persons (State)	0TPLA_0IDM_IS05_Q0004_V01		
Roles with Their System Access and Privileges (State)	0TPLA_0IDM_IS06_Q0001_V01	IDM Role Privilege Assignment	Role Privilege
Systems with Their Privileges and Assigned Roles (State)	0TPLA_0IDM_IS06_Q0002_V01		Role-Privilege Assignment
Roles with Their System Access and Privileges (History)	0TPLA_0IDM_IS06_Q0003_V01		
Systems with Their Privileges and Assigned Roles (History)	0TPLA_0IDM_IS06_Q0004_V01		

Note

The technical name for each corresponding query can be derived from the technical name from the Web template. The syntax is `OTPLA_<Technical_Name_of_Query>_V01`.

Note the following when executing the reports from the Web templates:

- The identity store to use for the reports is specified in the variable (`<OP_IDM_IS_M>`). If you want to specify a default identity store to use, see [Specifying the Default Identity Store to Use for Queries \[page 813\]](#). The end user can also specify a default identity store to use in the personalization options. He or she can also hide the *Identity Store ID* field if it always contains the same value.
- Use the *Person Selection by Attributes in Input Help* field to search for an identity using attributes, for example, the first or last name for the person. After the search, the internal ID (MSKEY) for the identity is displayed in the field.
- Use the *... or Unique ID of Person* field to search for an identity using the unique ID (MSKEYVALUE). Searching by means of other attributes is not possible in this field.
- When searching for an identity using the MSKEYVALUE, the search is case-sensitive.

Example

To search for the user with the MSKEYVALUE UserA, you must use the search input **UserA**. Using **usera** as the search input will not return *UserA* as the result.

Procedure

- Execute the reports by calling the corresponding URL for the Web template. The URL to use is the following:
`<http/https>://<hostname>:<port>/irj/servlet/prt/portal/prtroot/pcd!
3aportal_content!2fcom.sap.pct!2fplatform_add_ons!2fcom.sap.ip.bi!2fiViews!
2fcom.sap.ip.bi.bex?BI_COMMAND-BI_COMMAND_TYPE=SET_TEMPLATE&BI_COMMAND-TEMPLATE=
<Web_Template_Technical_Name>.`

Related Information

You can also display the queries as a BEx Web Application iView into the portal. For more information, see [Creating BEx Web Application iViews](#)

7.2.5 Additional Tasks

Additional tasks that you may need to perform after the initial set up include:

- Checking the progress of the data transfer.
This applies to either initial or delta loads.

- Starting the process chain manually (if required).
- Specifying the default identity store to use for queries.
- Configuring the use of SSL for the connections.
- Checking the connection parameters to the BW system on the VDS.
- Customizing the attribute to use for mapping between user IDs in the identity store and the BW system.
- Release 7.1 only: Adjust the configuration after updating components.
- Troubleshooting typical error situations.

7.2.5.1 Checking the Progress of the Data Transfer

Context

In particular for the initial load, the data transfer may take some time, depending on the number of records being transferred. To check the progress of this data transfer, you must log on to the BW system and use the BW functions.

Procedure

1. In the Data Warehousing Workbench (transaction RSA1) on the BW system, navigate to the Web service DataSource for the InfoArea Identity Management. This is the DataSource *Identity Management Entries* with the technical name `0IDM_ENTRY_ATTR`.

Example

There are several ways to navigate to this object.

1. Select *Source Systems* in the left pane. Expand the *Web service* node and select the source system you created for identity management with a double-click (for example, *SAP ID Management Reporting*).
The DataSources for this Source System is displayed.
 2. Expand the DataSource tree until you locate the DataSource *Identity Management Entries*.
2. Select the DataSource *Identity Management Entries* with a double-click and choose  *Goto*  *GotoManage* .
 3. Select the request for which you want to check the progress for and choose  (PSA Maintenance).

Results

The number of records that have been transferred with this request are displayed.

Note

There is no automatic refresh of the progress. Therefore, to continuously observe the progress, you must exit the screen and reopen it.

7.2.5.2 Starting the Process Chain Manually

If you did not set up the Web service to trigger the process chain automatically, you must start it manually (or schedule it to run periodically in the BW system). This section describes how to start it manually.

Prerequisites

The data transfer from SAP Identity Management is completed.

Procedure

1. In the Data Warehousing Workbench (transaction RSA1), choose [Administration](#).
2. Select [Process Chains](#) in the left pane.
3. Expand the display component node without a name and select the process chain to start with a double-click.
4. Choose  [Execution](#)  [Execute Synchronously](#)  to start processing immediately. (If you want to schedule the process chain execution or run a simulation, choose one of the options, as appropriate.)

7.2.5.3 Reloading InfoObject Master Data

After activating fixes or updates for BW transformations for Identity Management, you may have to delete all master data from the affected InfoObjects. Then you must reload the master data from the DataStore Object that permanently stores all data received from the Identity Center. SAP support or SAP notes will help determine whether this is required for your system. This section describes the required steps.

Prerequisites

You have already transferred data from your Identity Center to the BW system.

Procedure

1. Call transaction SA38 and enter the program RSBCT_IDM_DEL_IOBJ_MASTER_DATA.
2. Run the program and wait until it has finished.
3. In the Data Warehousing Workbench (transaction RSA1), choose *Modeling*.
4. Expand ► *Identity Management* ► *Attribute* ► *Attribute (Attribute)* ► *Data Transfer Processes* ► *Full DTP: OIDM_ENAT -> OIDM_AT* ► and select the Data Transfer Process with a double-click.
5. Choose ► *Execute* ► *Execute* ► to start the Data Transfer Process.
6. When asked whether to display the request monitor, choose *Yes*. Choose *Refresh* until the Data Transfer Process has been completed. Then select *Back* to return to the Data Warehousing Workbench.
7. Run the process chain manually as described in [Starting the Process Chain Manually \[page 812\]](#).

7.2.5.4 Specifying the Default Identity Store to Use for Queries

If you usually run the identity-related BW queries for the same identity store, you can maintain the ID of the identity store as a default value in the query variable **<Identity Store ID>** (0P_IDM_IS_M). You can set this variable in any of the identity management queries and the default value will apply in all queries that use it.

Prerequisites

You are familiar with the Business Explorer (BEx) Query Designer.

Procedure

1. Start the BEx Query Designer.
2. Open any of the identity management queries.

Note

For a list of the SAP-delivered identity management queries, see [Using the Web Templates \[page 808\]](#).

3. Select the *Filter* tab page.
4. Navigate to one of the *Identity Store ID* characteristics and choose *Restrict...* from the context menu.
5. In the *Show* field, select *Variables* from the dropdown box.
6. Select the **<Identity Store ID>** variable (not the **<Identity Store ID>** (<Copying Value of 0P_IDM_I_M> variable) and choose the symbol for *Edit Variable*.
7. Choose the *Default Values* tab.
8. Choose *Change Default Values*.

9. In the *Show* field, select *Single Values*.
10. Select the identity store ID to use as the default and choose *OK* for each of the various screens that you navigated through.
11. Save the query.

Note

Users can also save a default value as a variant in their personalization options.

7.2.5.5 Configuring the Use of SSL for the Connections

There are two connections that can be secured with SSL:

- The Web service connection between the VDS and the BW system
- The LDAP connection between the Identity Center and the VDS

See the sections that follow.

7.2.5.5.1 Configuring SSL for the Web Service Call from the VDS to the BW System

Prerequisites

- You have access to the certificate to use to establish trust with the BW server. This is either the SSL server certificate from the BW server itself (in the case of a self-signed certificate), or it is the root certificate from the Certification Authority (CA) that issued the BW server its certificate.
- You know the location of the keystore file to use for the truststore. By default, this is the file `cacert.ts` located at `<Path_to_Java_Directory>\lib\security\cacert.ts`. You can check the path to the Java directory in the VDS GUI under **Tools > Options**.

Procedure

- [Exporting the Certificate to Establish Trust \[page 815\]](#)
- [Creating the Keystore Reference to the Truststore on the VDS \[page 816\]](#)
- [Importing the Certificate to Use to Establish Trust \[page 817\]](#)
- [Creating a Keystore Reference and Key Pair for the VDS to Use for Mutual Authentication \(Optional\) \[page 817\]](#)

7.2.5.5.1.1 Exporting the Certificate to Establish Trust

Context

Use the trust manager (transaction STRUST) on the BW server to locate and export the certificate to use.

Procedure

1. In the BW system, start the trust manager (transaction STRUST).
2. Expand the *SSL server PSE* node and select a server with a double-click. The server's certificate appears in the upper section of the screen. If the certificate is a self-signed certificate, the text (*Self-Signed*) appears under the certificate information in the Own cert. field.

Note

Make sure this certificate matches the server you want to connect to with the BW connector. For example, if all BW servers use a certificate that has been issued by the same CA, then selecting any server suffices. However, if different application servers use different self-signed certificates, export the certificates belonging to all of the servers that the BW connector will connect to.

3. Select the certificate with a double-click.
The certificate details are displayed in the lower section of the screen. In this case of a self-signed certificate, export this certificate. Otherwise:
 1. Note the name of the issuer from the *Issuer* field.
 2. In the certificate section, choose the symbol for  (Import certificate).
 3. Select the *Database* tab page.
 4. Use the input help to search for and select the certificate that belongs to the issuer. Choose *Enter* to continue.
The CA root certificate appears in the lower section of the screen.
4. Export the certificate to a file by choosing the symbol for  (Export certificate).
5. Select the *File* tab page.
Use either format, but make sure the file has the extension *.der* when using the binary format or *.pem* for the base 64 format.
6. Note the location of the file. You will need it to import the certificate into the VDS truststore.

Note

If the certificate was issued by a CA, but the CA's root certificate is not located in the trust manger's certificate database, then you must locate the CA root certificate by other means. Many CA root certificates are available in the most widely used Web browsers or you have to obtain it from the CA.

7.2.5.5.1.2 Creating the Keystore Reference to the Truststore on the VDS

The truststore contains the certificates belonging to the communication partners that are trusted, or those belonging to the issuing CAs. The default truststore used by the Java environment is `cacerts`, and although you can create and use a different truststore, the description below applies to using the default `cacerts` truststore.

Prerequisites

You know the location of the `cacerts` keystore file in the file system. You also know the password needed to access it.

Context

To create the keystore reference to the truststore:

Procedure

1. Start the VDS.
2. If the BW connector configuration is not already open, open it.
3. Expand the *Virtual Directory Server* configuration.
4. Select the *Keystore references* node and choose *New* from the context menu (right mouse button).
5. Choose *Select ...* and navigate to the location in the file system where the file `cacerts` is located and enter the password. Choose *OK* to return to the BW connector configuration.

Note

If you specified the path and password when configuring the BW connector, then use the information you provided then for this step. Otherwise, you must also manually enter the path and password in the corresponding constants for the BW connector. The parameters to maintain are:

- `MX_TRUSTSTORE_PATH`
- `MX_TRUSTSTORE_PWD`
- `MX_KEYSTORE_PATH`
- `MX_KEYSTORE_PWD`

See also [Checking the Connection Parameters to the BW System on the VDS \[page 821\]](#).

7.2.5.5.1.3 Importing the Certificate to Use to Establish Trust

Procedure

1. Choose [Show contents...](#) from the context menu for the keystore reference to the truststore (for example, cacerts).
2. Choose [Add Certificate](#).
Select the certificate to import from the corresponding location in the file system and choose [Open](#).
The certificate is displayed.
3. Choose [OK](#) to insert the certificate into the keystore.
4. Enter a name to use for the entry and choose [OK](#).

7.2.5.5.1.4 Creating a Keystore Reference and Key Pair for the VDS to Use for Mutual Authentication (Optional)

Context

If the VDS is to use a client certificate for authentication on the BW server, then create a key pair and certificate to use for authentication.

Proceed as follows:

Procedure

1. Create a keystore reference for the key pair to use, for example VDS_Client.
2. Choose [Show contents...](#) from the context menu for this keystore reference.
3. Choose [Add New Self-Signed Certificate](#).

Note

This step creates a self-signed certificate for the VDS. If you want to have the certificate signed by a Certification Authority (CA), then you can use the *keytool* utility to create the certificate request and import the corresponding certificate request. For more information, see the document *Using SSL for LDAP communication*.

4. Enter the data and choose [OK](#).
5. Export this certificate to the file system by choosing [View certificate](#).
6. Choose [Copy to file](#) and save the certificate using either format.
7. Return to the BW system and import this certificate into the SSL server PSE:

1. Return to the trust manager.
2. In the upper section of the screen, make sure the SSL server PSE is selected and displayed.
3. In the lower section of the trust manager screen, choose the symbol for  (Import certificate).
4. Select the format to use (either Binary for a certificate that was saved using the **.der** format, or Base 64 for one saved using the **.pem** format) and choose *Enter*.
5. Choose the button for  (Add to Certificate List).
6. Save the data.
8. Using table view maintenance (transaction **SM30**), maintain the table view VUSREXTID for the **<External ID type> = DN**. In this table view, specify the user mapping between the Distinguished Name found in the certificate and the user ID for the user used for the connection.

7.2.5.5.2 Configuring SSL for the Connection from the Identity Center to the VDS

Context

To use SSL between the Identity Center and the VDS, the VDS must possess an SSL server certificate that is trusted by the Identity Center. The procedures below describe how to acquire the certificate and establish trust between the two components.

Procedure

- [Creating the Keystore Reference on the VDS \[page 818\]](#)
- [Creating a Key Pair for the VDS and Exporting the Public-Key Certificate \[page 819\]](#)
- [Importing the Certificate into the Truststore Used by the Identity Center \[page 819\]](#)
- [Enabling SSL on the VDS \[page 820\]](#)
- [Configuring the Jobs to Use SSL \[page 820\]](#)

7.2.5.5.2.1 Creating the Keystore Reference on the VDS

Context

Create a keystore reference to use for the SSL server certificate and generate the corresponding key pair.

Procedure

1. Start the VDS.
2. If the BW connector configuration is not already open, open it.
3. Expand the *Virtual Directory Server* configuration.
4. Select the *Keystore references* node and choose *New* from the context menu (right mouse button).
5. Enter a display name for the keystore reference, for example, **VDS_Server** and choose *Create ...* to create a new keystore entry. (Choose *Select...* if the file already exists.)
6. Select the location in the file system to store the entry and enter the password.
7. Choose *OK* to return to the BW connector configuration.

7.2.5.5.2.2 Creating a Key Pair for the VDS and Exporting the Public-Key Certificate

Procedure

1. To create a key pair to use, choose *Show contents...* from the context menu for the keystore reference.
2. Choose *Add New Self-Signed Certificate*.

Note

This step creates a self-signed certificate for the VDS. If you want to have the certificate signed by a Certification Authority (CA), then you can use the *keytool* utility to create the certificate request and import the corresponding certificate request. For more information, see the document *Using SSL for LDAP communication*.

3. Enter the data and choose *OK*.
4. Export this certificate to the file system by selecting the certificate and choosing *View certificate*.
5. Choose *Copy to file* and save the certificate using either format. Note the location of the certificate. You will need it to import it into the truststore used by the Identity Center.

7.2.5.5.2.3 Importing the Certificate into the Truststore Used by the Identity Center

Prerequisites

You know how the Java environment is set up so you can determine the location of the truststore file (cacerts). You can check the path to the Java VM in the Identity Center console under ► *Tools* ► *Options* ►

Context

To import the VDS server certificate into this keystore, you can use the keystore maintenance tool on the VDS. Create or open the keystore and import the certificate from the VDS into it as follows:

Procedure

1. If it does not already exist, create a keystore reference to the `cacerts` truststore. For more information, see [Configuring SSL for the Web Service Call from the VDS to the BW System \[page 814\]](#) in the section *Creating the Truststore Reference on the VDS*.

Note

If the VDS and the Identity Center are installed on the same machine and share the same Java environment, they can also use the same `cacerts` file as the truststore.

2. Choose [Show contents...](#) from the context menu for this reference.
3. Choose [Add Certificate](#) and load the file you saved above from the file system into the truststore.

7.2.5.5.2.4 Enabling SSL on the VDS

Procedure

1. If the VDS is running, stop it.
2. Expand [Deployments](#) > [LDAP Deployments](#) > [main_listener](#).
3. Select the [main_listener](#) with a double-click.
4. Enter the SSL port number. The default port for SSL with LDAP is 636.
5. Select the [Enable secure server](#) option.
6. Select the keystore that contains the SSL server certificate from the dropdown box, for example, [VDS_Server](#).
7. Choose [Apply](#).
8. Restart the listener.

7.2.5.5.2.5 Configuring the Jobs to Use SSL

Context

In the Identity Center, set up the data transfer job to use SSL for the connection.

Procedure

1. Modify the repository constant `VDS2BW_PORT` for the repository that is used for the data transfer to the VDS. Set it to the HTTPS port (default=636).
2. Adjust the passes in the *IDM to BW Data Transfer* job. For each *To LDAP Directory* pass do the following:
 1. Choose the *Destination* tab page.
 2. Select *SSL* as the *Security option*.
 3. Choose *Apply*.

7.2.5.6 Checking the Connection Parameters to the BW System on the VDS

The connection parameters to the BW system on the VDS are stored in constants. To check or change these values, expand *Virtual Directory Server* in the BW connector configuration and select *Constants* with a double-click. The table below shows the constants to maintain.

BW Connection Parameter Constants

Constant	Description	Value
<code>MX_HOST</code>	Hostname of the BW server.	<code><hostname></code>
<code>MX_PORT</code>	Port to use for the connection.	<code><port></code>
<code>MX_SAP_CLIENT</code>	SAP client of the BW system.	<code><SAP client></code>
<code>MX_USERNAME</code>	User ID to use for authentication with user ID and password.	<code><user_ID></code>
<code>MX_PASSWORD</code>	Password to use for authentication with user ID and password.	<code><password></code>
<code>MX_COMM_SEC</code>	Flag indicating whether HTTP or HTTPS is to be used for the connection.	<code><1,2></code> <code>1 = HTTP, 2 = HTTPS</code>
<code>MX_AUTH_METH</code>	Authentication method to use for the connection.	<code><1,2></code> <code>1 = user ID and password, 2 = SSL with mutual authentication</code>
<code>MX_TRUSTSTORE_PATH</code>	Path to the truststore that contains the certificate to use to establish trust to the BW server.	<code><Path to truststore> (cacerts)</code>
<code>MX_TRUSTSTORE_PWD</code>	Password to use to access the truststore that contains the certificate to use to establish trust to the BW server.	<code><Path to truststore> (cacerts)</code>
<code>MX_KEYSTORE_PATH</code>	Path to the keystore that contains the key pair and public-key certificate belonging to the VDS.	<code><Path to keystore></code>

Constant	Description	Value
MX_KEYSTORE_PWD	Password to use to access the keystore that contains the key pair and public-key certificate belonging to the VDS.	<Password for keystore>
MX_TRANSFER_SERVICE	Web service of the BW system that accepts data from SAP Identity Management.	sap/bc/srt/rfc/BI0/CFIDM_002
002MX_TRANSFER_SERVICE_ENDPOINT_FALLBACK	Fallback for data transfer Web Service endpoint.	Blank unless inconsistencies exist. See the note at the end of this section.
MX_CHAIN_START_URL_1	Needed to compile Process Chain Web Service Endpoint URL.	sap/bc/srt/rfc/sap
MX_CHAIN_START_URL_2	Needed to compile Process Chain Web Service Endpoint URL	rs_bct_idm_chain_start
MX_CHAIN_START_HTTP_ENDPOINT	Needed to compile Process Chain Web Service Endpoint URL	default
MX_CHAIN_START_HTTPS_ENDPOINT	Needed to compile Process Chain Web Service Endpoint URL	ssl
URLssIMX_CHAIN_START_SERVICE_ENDPOINT_FALLBACK	Fallback for Process Chain Web Service Endpoint	Blank unless inconsistencies exist. See the note at the end of this section.
MX_WSENDPOINT (Release 7.1 SPS 5 up to patch 1 only)	Path to the end point for the Web service used to process the initial or delta loads.	sap/bc/srt/rfc/BI0/CFIDM_002
MX_WSCHAINSTARTENDPOINT (Release 7.1 SPS 5 up to patch 1 only)	Path to the end point for the Web service that triggers the process chain.	sap/bc/srt/scs/rs_bct_idm_chain_start, if simplified Web service configuration was used. Otherwise sap/bc/srt/rfc/sap/rs_bct_idm_chain_start/<client>/rs_bct_idm_chain_start/<default/ssl>. See the note below.

The endpoint used for the Web service that triggers the process chain is built from the constants mentioned as follows: <MX_CHAIN_START_URL_1> + /<MX_CHAIN_START_URL_2> + /<MX_SAP_CLIENT> + /<MX_CHAIN_START_URL_2> + /<MX_CHAIN_START_HTTPS_ENDPOINT> or <MX_CHAIN_START_HTTP_ENDPOINT>.

Note

In SAP Identity Management Release 7.1 SPS 5, it is specified directly in MX_WSCHAINSTARTENDPOINT.

Example

An example of an HTTP endpoint is sap/bc/srt/rfc/sap/rs_bct_idm_chain_start/<SAP_client>/rs_bct_idm_chain_start/default.

To verify that the endpoint corresponds to the settings in the BW system, see [Checking the Endpoint to Use for the Web Service that Triggers the Process Chain \[page 823\]](#).

7.2.5.6.1 Checking the Endpoint to Use for the Web Service that Triggers the Process Chain

Prerequisites

You have the authorization to use the SOA Manager. The corresponding ABAP role is SAP_BC_WEBSERVICE_ADMIN_BIZ.

Procedure

1. In the BW system, start the SOA Manager (transaction SOAMANAGER).
The *SOA Management* application starts in a Web browser.
2. Navigate to the service administration. As of SAP BW Release 7.3: ► *Service Administration* ► *Web Service Configuration* ►
The screen element names vary according to release. Some interfaces also display the terms in brackets.
3. Search for the *Service (Definition)* RS_BCT_IDM_CHAIN_START.
4. Select this service from the search results and choose *Apply Selection*.
The details for the Web service RS_BCT_IDM_CHAIN_START are displayed.
5. Choose the *Overview* tab.
6. In the dropdown list at the bottom of the screen, select the *RS_BCT_IDM_CHAIN_START::default* binding.
7. Choose *Open WSDL document for selected binding (or service)*.
The WSDL document opens.
8. Search for *wSDL:service name="RS_BCT_IDM_CHAIN_START"*. The endpoint is specified in the *soap:address location*. It uses the syntax `http://<hostname>:<port>/<endpoint>`.

❁ Example

For the address location `http://host123.mycompany.com:50000/sap/bc/srt/rfc/sap/rs_bct_idm_chain_start/999/rs_bct_idm_chain_start/default`, the endpoint is `sap/bc/srt/rfc/sap/rs_bct_idm_chain_start/999/rs_bct_idm_chain_start/default`.

📘 Note

For HTTPS endpoints, use *ssl* instead of *default*.

📘 Note

If, for some reason, the Web service URLs for your BW system do not match the described pattern, you can store the complete URL in the constant MX_CHAIN_START_SERVICE_ENDPOINT_FALLBACK (for the Web service that triggers the process chain) or MX_TRANSFER_SERVICE_ENDPOINT_FALLBACK FALLBACK (for the Web service that accepts data from SAP Identity Management. If specified, these URLs will be used instead of building the complete URL from the values specified in the other constants.

7.2.5.7 Customizing the Attribute to Use for Mapping Between User IDs in the Identity Store and in BW

To determine the correct authorizations (that is, manager or owner) when using the Web templates, the BW system needs to know which user ID in the BW system maps to the user ID in the identity store. By default, the attribute used for this mapping is the MSKEYVALUE for an identity, and the corresponding transformation in BW assumes that a user with this ID exists in the BW system. If you want to use a different attribute to specify the BW user ID, you must specify the name of the attribute to use.

Procedure

1. In the Data Warehouse Workbench, navigate to the process chain. (In the left area of the screen, choose [Administration](#) and then select [Process Chains](#).)
2. Expand the display component node without a name and select the process chain [Identity Management: Delta Load](#) with a double-click.
The process chain is displayed in the right section of the screen.
3. Locate the step [Full DTP: OIDM_PNAT -> OIDM_USPN](#).
4. From the context menu for this step, choose [Target Administration](#).
The [InfoProvider Administration](#) screen appears.
5. Choose [Edit](#) [Change Update Rule](#).
The [Transformation Display](#) screen appears.
6. Change to edit mode by choosing  (Display <—> Change).
7. Choose [Start Routine](#).
8. Edit the start routine method. Specify the attribute to use for the mapping in the Customizing section.

```
*$*$ begin of routine - insert your code only below this line *-*
" Keep only the relevant records in the source package.
DELETE SOURCE_PACKAGE
WHERE IDM_ATNAME <>
" Customizing: Maintain the name of the attribute that contains
" the person's ABAP user ID in this BW system.
<Attribute_Name>.
*$*$ end of routine - insert your code only before this line *-*
```

7.2.5.8 Typical Error Situations

The table below shows the most common sources of errors that can occur when setting up and running the BW connector with SAP Identity Management.

Typical Error Situations

Problem	Symptom	Solution
Connection between Identity Center and VDS fails.	Error message in job log: <code>javax.naming.CommunicationException: <hostname>:<port> [Root exception is java.net.ConnectException: Connection refused: connect]</code>	Make sure the VDS is running. Check the connection parameters (hostname of VDS, port, user ID, and password). These are specified in the job constants for the initial load and delta jobs.
	Error message in job log: <code>javax.naming.AuthenticationException: [LDAP: error code 49 - Explicit authentication failed]</code>	Check the authentication information used for the connection (user ID and password). These are specified in the job constants for the initial load and delta jobs.
You are using SSL for the connection between the Identity Center and the VDS, and this connection fails.	Error message in job log: <code>javax.naming.CommunicationException: <hostname>:<SSL_port> [Root exception is java.net.ConnectException: Connection refused: connect]</code>	Make sure the VDS is running as a secure server. In addition, make sure the SSL port is used for the connection. For more information, see Configuring SSL for the Connection from the Identity Center to the VDS [page 818] in the section <i>Enabling SSL on the VDS</i> .
	<code>javax.naming.CommunicationException: simple bind failed: localhost:636 [Root exception is javax.net.ssl.SSLHandshakeException: sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilder Exception: unable to find valid certification path to requested target]</code>	Make sure the VDS possesses an SSL server certificate. Make sure the truststore <code>cacerts</code> on the Identity Center's server contains the SSL server certificate from the VDS. For more information, see Configuring SSL for the Connection from the Identity Center to the VDS [page 818] in the sections <i>Creating a Key Pair and Exporting the Public-Key Certificate</i> and <i>Importing the Certificate into the Truststore</i> .
	Error message in job log: <code>javax.naming.ServiceUnavailableException: localhost:636; socket closed</code>	Make sure SSL is enabled in the job passes. For more information, see Configuring SSL for the Connection from the Identity Center to the VDS [page 818] in the section <i>Configuring the Jobs to Use SSL</i> .

Problem	Symptom	Solution
Connection from VDS to the BW system fails.	Error message in job log: Exception from Add operation:javax.naming. NamingException: [LDAP: error code 1 - (WS Client Data Source:1:: nested exception is: java.net.ConnectException: Connection refused: connect)]; remaining name 'cn=EOT, o=bwconnector'	Check the connection parameters to the BW system (hostname of the BW system, client, and port).For more information, see Checking the Connection Parameters to the BW System on the VDS [page 821] .
	Error message in job log: Exception from Add operation:javax.naming. NamingException: [LDAP: error code 1 - (WS Client Data Source:1:(401)Unauthorized)]; remaining name 'cn=EOT,o=bwconnector'	Check authentication information used for the connection (user ID and password).For more information, see Checking the Connection Parameters to the BW System on the VDS [page 821] .
You are using SSL for the connection between the VDS and the BW server, and this connection fails.	Error message in job log: Exception from Add operation:javax.naming. NamingException: [LDAP: error code 1 - (WS Client Data Source:1:: nested exception is: java.net.ConnectException: Connection refused: connect)]; remaining name 'cn=EOT, o=bwconnector'	Check the connection parameters to the BW system (hostname of the BW system, client, and port). For more information, see Checking the Connection Parameters to the BW System on the VDS [page 821] . Make sure that SSL is active on the BW server.For more information, see the AS ABAP documentation.
	Error message in job log: Exception from Add operation:javax.naming. NamingException: [LDAP: error code 1 - (WS Client Data Source:1:: nested exception is: javax.net.ssl.SSLHandshakeException: sun. security.validator.ValidatorException: No trusted certificate found)]; remaining name 'cn=EOT,o=bwconnector'	Make sure the truststore cacerts on the VDS server contains the SSL server certificate to trust from the BW system. For more information, see Configuring SSL for the Web Service Call from the VDS to the BW System [page 814] .
Process chain does not start	startError message in job log: Exception from Add operation:javax.naming. NamingException: [LDAP: error code 1 - (WS Client Data Source:1:(500)An exception with the type CX_SY_RANGE_OUT_OF_BOUNDS occurred, but was neither handled locally, nor declared in a RAISING clause)]; remaining name 'cn=EOT,o=bwconnector'	Make sure the Web service is configured to trigger the process chain automatically. For more information, see Configuring the Web Service to Trigger the Process Chain [page 798] . Make sure the endpoint for the Web service is specified correctly in the BW connector constants. For more information, see Checking the Connection Parameters to the BW System on the VDS [page 821] .

Problem	Symptom	Solution
	No error message occurred.	Make sure the <i>End of Transmission</i> job pass destination contains the flag that triggers the process chain. <i>StartChain=D</i>

8 SAP Identity Management REST Interface Version 2

Version 2 of Identity Management REST (Representational state transfer) service implements the Open Data Protocol (OData) in version 2.0.

Consumers of this service API (Application Programming Interface) should be familiar with OData before implementing their own applications. For more information about the Open Data protocol, see the OData web site.

Version 2 of the Identity Management REST service supports - as does OData - both formats for representing the resources it exposes, the XML-based Atom format and the JavaScript Object Notation (JSON) format. In contrast, version 1 of the Identity Management REST service only supported the JSON format. For readability's sake, all examples in this document are based on the JSON format, with the exception of the service metadata document that is only returned in XML format.

OData is designed to be modular. This means, that the Identity Management REST service only implements as much of the OData specification as required for the intended target scenarios and uses the Apache Olingo open source library. Furthermore, the support for Query Options is limited in version 2. To allow the implementation of further features, for example, to support more Query Options, within the REST interface in later service packs, clients can call a service operation that returns version information. Clients can then check whether certain features can be used or not.

The Identity Management REST interface supports SAP NetWeaver 7.3 SP09, SAP NetWeaver 7.31 SP06 and higher, SAP NetWeaver 7.4 SP02 and higher and SAP NetWeaver 7.5 SP08 and higher as runtime environment and requires SAP Identity Management 8.0 or higher. The interface is contained in the `IDMREST<IdM SP version>_<IdM Patch version>.sca`.

It is still possible to use the version 1 of the Identity Management REST service. However, the Identity Management REST Interface Version 2 has clear advantages over its previous version, and is the recommended version of the Identity Management REST service. However, be aware that the Identity Management REST API version 2 supports only Entry types which are marked as **Searchable entry type**. For more information, see [General \(Entry Type\) \[page 981\]](#).

Related Information

[Open Data Protocol](#) ➡

[SAP NetWeaver Identity Management Identity Center REST API](#) ➡

8.1 Security

This section covers the following security topics for the REST Interface:

- Authentication and authorization
 - Necessary UME actions
 - Better performance with single sign-on with logon tickets
- Cross-site request forgery protection

⚠ Caution

As of SAP Identity Management 8.0 SP06 and higher, Identity Management REST Interface v2 requires HTTPS communication by default. For more information, see [Configuring the Java System Properties](#)

Related Information

[Authentication and Authorization \[page 829\]](#)

[Cross-Site Request Forgery Protection \[page 830\]](#)

8.1.1 Authentication and Authorization

Most calls to the provided service URIs (Uniform Resource Identifiers) require a user to authenticate. The user must have the UME actions `idm_authenticated` and the `idm_authenticated_restapi`.

As the REST service is an application deployed on the SAP NetWeaver AS for Java, you can also use logon ticket authentication instead of basic authentication to improve performance.

Related Information

[Configuring the UME Actions for the REST Interface \[page 842\]](#)

[Configuring Single Sign-On With Logon Tickets in the REST Interface for AS Java 7.1 and higher \[page 842\]](#)

8.1.1.1 Single Sign-On With Logon Tickets

The default configuration of the SAP NetWeaver Identity Management 7.2 REST interface forces a logon on all requests using the provided basic authentication credentials. This authentication consumes time and leads to a high number of security sessions in the SAP NetWeaver AS for Java.

The system has to maintain these sessions until they time out. This impacts the performance of the REST interface for subsequent requests.

To improve performance, configure single sign-on (SSO) with logon tickets for the REST interface.

Related Information

[Configuring Single Sign-On With Logon Tickets in the REST Interface for AS Java 7.1 and higher \[page 842\]](#)

8.1.2 Cross-Site Request Forgery Protection

The Identity Management REST service protects against Cross-Site Request Forgery (CXRF/XSRF) attacks.

Cross-Site Request Forgery is an attack that tricks a victim's browser into sending a request to a vulnerable web application, which then performs an undesired action on behalf of the victim (for example, changing credentials, making an illegal purchase, or performing online financial operation). This vulnerability is extremely widespread since the majority of today's web applications rely solely on automatically submitted credentials such as session cookies, basic authentication credentials, source IP addresses, Secure Socket Layer (SSL) certificates, or Microsoft Windows domain credentials. Therefore, as the user is currently authenticated to the site, the site will have no way to distinguish the XSRF attack from a legitimate user request.

To protect against attacks of this type, the Identity Management REST interface uses a standard filter provided by the SAP NetWeaver Application Server for Java (AS Java).

- The XSRF protection for the SAP NetWeaver REST interface is based on an XSRF token and applies to all modifying requests. The protection works as follows:
 - With a non-modifying request, authenticate to the server and retrieve an XSRF token.
The client sends a non-modifying request (GET, HEAD, OPTIONS), where the **X-CSRF-Token** header field has the value **Fetch**. This request also includes the regular authentication information required by the REST interface, like user ID and password, logon ticket. The SAP Identity Management REST interface generates an XSRF token and sends it back in the **X-CSRF-Token** header of the response. This token is associated with the session established between the client and the server.

Note

The response of a failed request does not contain a valid XSRF token.

- With each modifying request, include the XSRF token.
The client sends a modifying request, where the same request header (**X-CSRF-Token**) has the value of the previously received token. The server validates the XSRF token. Additionally, the request includes

authentication data (as far as required again, depending on the login module stack used for the REST interface) and especially all cookies set by the server during the non-modifying request. The cookies include session information, which is required by the server to verify the XSRF token because of its association with the current session. If the validation fails, the server replies with an error response with HTTPS status code 403 (Forbidden).

Note

If two services are in the same security policy domain, authenticating with the first service allows the user to access the second one without additional authentication. Also, the received XSRF token is valid for all services across the security policy domain.

Related Information

[Authentication and Authorization \[page 829\]](#)

[XSRF Protection for REST Services](#)

[Specifying Authentication Mechanisms for Java Applications \(Description for setting the security policy domain.\)](#)

[SAP Identity Management Identity Center REST API](#) 

[Virus Scanning \[page 896\]](#)

8.2 Content Type Negotiation

The REST service can use the existing HTTPS headers to indicate or negotiate the representation of the content type of a request or response. Additionally, it allows the usage of an OData System Query Option to request a content type of a response.

8.2.1 Content Type Via Request Header

The REST client can specify which representation of the resource it prefers by including the HTTPS request header “accept” with a list of MIME types it can handle.

If the header is not specified, the Atom representation is returned by default. If the requested MIME type is not supported by the REST service, the error code 406 (**Not Acceptable**) is returned. When a resource is sent as part of the request or response body, the client and server have to indicate the corresponding content type in the header `content-type`.

Example

HTTPS GET Request

Accept: application/json

Example

HTTPS POST Request

Accept: application/json

Content-Type: application/json

8.2.2 Content Type Via System Query Option

Using the Format System Query Option (`$format`), the REST client can specify which representation of the resource the response returns. If a request URI contains this option, it overrides the value specified in the accept request header.

Syntax

`formatQueryOp = "$format=" ("json" / "atom")`

Example

HTTPS Request

GET `./idmrestapi/v2/service?$format=json` HTTPS/1.1

8.2.3 Character Encoding

The expected character encoding when sending requests to the REST interface is UTF-8.

The used character encoding is specified in the HTTPS response header `Content - Type`. Currently this is always UTF-8.

Example

HTTPS Request

```
Charset: UTF-8
```

Example

HTTPS Response

```
Content-Type: application/json; charset=UTF-8
```

8.3 Localization

The REST interface supports the retrieval of localized texts for which translated texts exist. This applies, for example, to display names of attributes or localized value help texts contained in attributes.

If the locale of the corresponding step is not specified or not maintained in Identity Management, the locale is determined in the following order. This order is similar to the one described for the user management engine (UME) in *User Profile*: "If a language is not explicitly defined for a user in the user profile, then the browser language is used. If the browser language is not set, then the server's default language is used."

Order for determining the language:

1. The REST client can send the preferred language by sending the **Accept - Language** HTTPS request header.
2. The language of the logged-in user.
3. The default locale of the Java Virtual Machine (JVM).
4. English is used as the default language.

Example

HTTPS Request

Accept-Language: en

Related Information

[User Profile](#)

8.4 Performance

Context

This section covers the following performance topics for the REST Interface:

- OData Means
- Response Properties

8.4.1 OData Means

Following the OData specification, the REST service by default does not return the full graph of entries related to an entry or collection of entries requested by a URI.

It is deferred unless the client explicitly asks for this information using the **\$expand** system query option for these related entries. This conserves resources if this information is not needed, but also allows getting all information about related entries in one request if it is required by the client.

→ Tip

If a scenario always requires additional information that is deferred by default, we recommend that you use the **\$expand** option (if available), instead of always retrieving the information with additional REST calls to the "deferred content".

ⓘ Note

If too many entries exist, retrieving a collection of entries (for example, executing a search for entries) may return a partial result list. Partial result lists are marked with a **__next** property that contains the link to the next partial set of entries.

In case you want to retrieve the count of the requested entries, use the **\$count** system query option. For example, the following request:

```
[GET]/idmrestapi/v2/service/TaskCollection/$count
```

returns the count of the approval items assigned to the logged-in user:

```
2
```

Related Information

[Expand System Query Option](#) ➔

8.4.2 Response Properties

According to the OData standard the REST service by default always delivers all properties defined in the Service Metadata Document for an entity at all operations that return an entity or entity set.

As the amount of properties for an entry type usually is quite large, the HTTPS response is also large. Consequently, the response time is rather long. But most clients are not interested in all these properties, especially as only for the properties that are exposed by the specified task values are returned. For all other properties the initial values are set.

Therefore, the Identity Management OData Service provides a custom query option. Using this option delivers only those single value attributes in a response that contain a value and are exposed by the specified task. This custom query option works for all operations that retrieve an entity or entity set.

Request Parameters

```
Parameter = 'selectSetSVAttributes' [ '=true' ]
```

Request Example 1

Return all single value attributes of the person with the given ID that are exposed by the given task and contain a value.

```
GET/idmrestapi/v2/service/ET_MX_PERSON(ID=7894,TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE')?selectSetSVAttributes
```

8.5 Comparison to the Previous Version of the Identity Management REST Interface

Version 2 of the Identity Management REST service has clear advantages over its previous version, and is the recommended version of the Identity Management REST service.

It is still possible to use the version 1 of the Identity Management REST service, but you should consider upgrading to SAP Identity Management REST Interface Version 2.

Note

Upgrading to the Identity Management REST Interface Version 2 from a previous version will involve a reconfiguration (reimplementation) of your existing solution.

8.5.1 URIs

The URIs for the different entities have changed slightly as compared to the first version of the REST interface.

For example, accessing an element of a list used to be done like this:

```
list/<element1>.
```

With version 2 of the interface an element is accessed by putting the value in brackets:

```
list(<element1>).
```

All items of the list are accessed in the same way as in the first version, for example:

```
list.
```

Note

In accordance with the OData specification all URIs in version 2 of the OData Rest interface are case-sensitive.

Example Version 1

```
https://<host>:<port>/idmrest/v72alpha/entries/190/tasks/
```

Example Version 2

```
https://<host>:<port>/idmrestapi/v2/service/  
ET_MX_PERSON(ID=190, TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE')
```

8.5.2 Operations

The first version of the REST interface only supports the HTTPS methods **GET** and **POST** for managing all supported operations (retrieve, create and update) on the exposed entries.

Version 2 of the REST interface supports all HTTPS methods that are supported by the OData service interface, except the **MERGE** HTTPS method and **DELETE** HTTPS method. Instead, method tunneling must be done via **POST** HTTPS method including the header "X-HTTP-Method" with the value **MERGE**.

The HTTPS methods have a uniform meaning over all the entities it can handle.

Operation	HTTPS Method
Retrieve an entity or a set of entities	GET
Create a new entity	POST
Update an entity (via merge)	POST with MERGE method tunneling

8.5.3 Return Format

While the return format of the first interface version was JSON-only, version 2 of the REST interface also supports the Atom format.

The JSON response and the data structure within version 2 of the REST interface are adapted to be OData compliant. "In responses payloads (not request payloads) all the JSON representations [...] are wrapped by an outer most object that includes a single name/value pair. The name of the name/value pair is always "d" and the value is the JSON representation of an OData resource [...]. This pattern ensures JSON payloads returned from OData services are valid JSON statements, but not valid JavaScript statements. This prevents an OData JSON response from being executed as the result of a cross site scripting (XSS) attack."

Example Version 1

```
{
  "MX_REST_SUCCESS": true,
  "ENTRIES": {
    "MX_MAIL_PRIMARY": {
      "DISPLAYVALUE": "max.mustermann@example.com",
      "VALUE": "max.mustermann@example.com"
    },
    "MX_FIRSTNAME": {
      "DISPLAYVALUE": "Max",
      "VALUE": "Max"
    },
    "MX_PHONE_PRIMARY": {
      "DISPLAYVALUE": "+49 6227 8-12345",
      "VALUE": "+49 6227 8-12345"
    },
    "MX_MOBILE_PRIMARY": {
      "DISPLAYVALUE": "+49 123 4567890",
      "VALUE": "+49 123 4567890"
    },
    "DISPLAYNAME": {
      "DISPLAYVALUE": "Max Mustermann",
      "VALUE": "Max Mustermann"
    },
    "MX_LASTNAME": {
      "DISPLAYVALUE": "Mustermann",
      "VALUE": "Mustermann"
    },
    "MSKEYVALUE": {
      "DISPLAYVALUE": "D0123456",
      "VALUE": "D0123456"
    }
  }
}
```

```
}
  }
}
```

Example Version 2

```
{
  "d": {
    "__metadata": {
      "uri": "/idmrestapi/v2/service/
ET_MX_PERSON(ID=190,TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE')",
      "type": "IdMRestModel.ET_MX_PERSON"
    },
    "ID": 190,
    "TASK_GUID": "35B264BF-A75A-447D-A4EA-7894725245CE",
    "SV_MSKEYVALUE": "D0123456",
    "SV_DISPLAYNAME": "Max Mustermann",
    "SV_MX_LASTNAME": "Mustermann",
    "SV_MX_FIRSTNAME": "Max",
    "SV_MX_MAIL_PRIMARY": "max.mustermann@example.com",
    "SV_MX_PHONE_PRIMARY": "+49 6227 8-12345",
    "SV_MX_MOBILE_PRIMARY": "+49 123 4567890",
  }
}
```

Related Information

[JSON Format](#) ➤

[Atom Format](#) ➤

8.5.4 Query Options

With the second version of the Identity Management REST interface new features like "expanding" are introduced.

For an overview of possible operators, see the OData web site.

Related Information

[Query String Options](#) ➤

8.6 Identity Store Support

Version 2 of the REST interface only supports the identity store specified in the Identity Management JMX application property `com.sap.idm.jmx.idstoreid`, as did the first version of the REST interface.

8.7 Limitations

This overview lists the limitations of the second version of the Identity Management REST service.

- Response Header
 - The response header does not contain the Content-Language in the HTTPS response header.
- If an error occurs, the error response always shows the language “en-US”, even if the error text is localized.
- Attributes with the data type “Audit flag”, “Task reference”, and “Attribute reference” are not supported and are not returned in an HTTPS response.
- Paging of search results is not supported yet by the Identity Management Rest Service. For more information, see *Search Entries*.
- The return value for attributes with non-text data types and the “hide input” presentation property is always null. For more information, see *Get Attribute Values of an Entry*.

Related Information

[Get Attribute Values of an Entry \[page 870\]](#)

[Search Entries \[page 851\]](#)

8.8 Compatibility

As the REST interface advances and different versions of the specification materialize, we need to define how the SAP Identity Management REST interface provider (provider, for short) and the SAP Identity Management REST interface consumers (consumers, for short) are supposed to interoperate. This section defines the compatibility rules that need to be adhered to.

Version Numbering

To properly communicate the idea of compatible and incompatible changes, compatible version increments MUST only be minor versions, not major versions. On the other hand, incompatible changes MUST require a new major version. As far as this document is concerned, the goal is to only extend compatibly, hence

introducing only minor versions. Features added through minor versions but the first non-beta version of a major release **MUST** be optional.

Each incompatible change to the OData service is reflected in a different major version. The version is embodied in the URI, leading to a new service endpoint with each such change. If exceptions are required, for example, for security reasons, they are explicitly communicated.

Rule 1: Introduction of Compatible Changes

Unless otherwise noted, each newer SAP Identity Management REST interface version is fully compatible with respect to previous versions. If a compatible change is introduced, a new minor version is required. This minor version number increment is reflected by the function import `GetVersion`, see *Get Version Information*.

The following changes are compatible (this list is very restrictive and is likely to be extended in the future):

- Adding new function imports (actions and functions), association sets, and entity sets to existing entity containers.
- Defining new associations, entity types, and complex types.
- Adding new properties or navigation properties to existing entity types if they meet the following requirements:
 - The properties are optional (nullable).
 - The navigation properties are optional.
- Adding new optional parameters to existing function imports, given that calling the new function import without the new parameter has the same meaning as calling the old function import for any given value combination for the existing parameters, for example, `f_new(x, null) == f_old(x)`.
- Defining new optional custom query options.
- Introducing paging support. Clients shall be prepared to support partial sets of entities as defined in the OData specification, see the OData web page.

It is important that consumers honor these rules appropriately. That means, consumers **MUST** expect the above changes to happen and **MUST NOT** rely on any technical means that would assume otherwise, for example, static proxy generation (a well-known flaw in the SOAP web service world) to consume content.

Rule 2: Introduction of Incompatible Changes

All changes not listed under rule 1 are incompatible.

Rule 3: Feature Querying and Default Values

Even for fully compatible evolutions, a consumer must know the features the provider supports. For major version changes a new endpoint and service URI are introduced. If the consumer is newer than the provider, it could, for example, try to invoke an action that is not implemented. To remedy this situation, consumers must be able to query the features the provider supports. OData service metadata documents provide the

right mechanism for this. Taking this technical detail as a basis, we follow the convention that the SAP Identity Management Rest interface exposes the features supported in the service metadata document. Additionally the consumer can call the `GetVersion` function import to retrieve the major and minor version to find out if a required feature is available.

Rule 4: Provider Backward Compatibility

The versioning scheme employed defines that the provider (which is bound to a particular endpoint and supports a specific minor version of a major version) is compatible to all lower minor versions of this major version. Beyond that, the provider is not forced to remain backward compatible. Specifically, new major versions are typically implemented in a separate provider bound to a separate endpoint.

Rule 5: Provider Forward Compatibility

The provider is not required to be forward compatible. That is, the provider is not required to be able to talk to consumers supporting only newer major versions of the SAP Identity Management REST interface. Within the boundaries of a major release, except for a set of (initially) defined mandatory features, new features added through minor versions **MUST** be optional, hence making it possible for the provider supporting a lower minor version of a specific major version to communicate with a consumer that supports a higher minor version of that same major version.

Rule 6: Consumer Backwards Compatibility

Consumers do not have to be backward compatible. That is, they do not have to support all (latest minor of all major) SAP Identity Management Rest interface versions – they can choose to support only one. Within the boundaries of a major version, however, the consumer **SHOULD** be able to deal with providers of a lower version (at the cost of a moderately reduced feature set).

Rule 7: Consumer Forward Compatibility

As outlined in rule 1, a consumer **MUST** be able to distinguish those pieces of a provider's response that it needs and disregard all those it does not understand. Note this notion of forward compatibility only applies within the boundaries of compatible versions of the SAP Identity Management REST interface. So as long as a consumer understands the SAP Identity Management REST interface 2.0 and the provider offers 2.1, everything should be fine. If the provider only offers 3.0, the consumer has no chance.

Consumers are advised to make use of the `$select` query option to avoid unnecessary data passing between provider and consumer.

Related Information

[Get Version Information \[page 949\]](#)

[OData web page](#) 

8.9 Configuring the UME Actions for the REST Interface

The necessary UME actions `idm_authenticated` and the `idm_authenticated_restapi` are provided as part of the software components containing the Identity Management web user interface and the REST service. To use the REST service and to access the Service Document and Service Metadata Document the user needs these UME actions.

In the UME administration, use the search term **tc~idm*** to find these actions. All other necessary authorizations for a service call are defined by the access control of the related Identity Management UI task.

Related Information

[Administration of Users and Roles in User Management Engine \(UME\) for SAP NetWeaver 7.3](#)

[Administration of Users and Roles in User Management Engine \(UME\) for SAP NetWeaver 7.3 EHP1](#)

[Administration of Users and Roles in User Management Engine \(UME\) for SAP NetWeaver 7.4](#)

[Authentication and Authorization \[page 829\]](#)

8.10 Configuring Single Sign-On With Logon Tickets in the REST Interface for AS Java 7.1 and higher

Context

Using SSO with logon tickets, each request sent to the REST interface without a logon ticket creates a logon ticket. The example application in SAP Note [1665777](#)  shows, how the logon tickets are read from the HTTP response and how they are attached to a subsequent HTTP request for the same user.

Note

The logon ticket contains authentication information of the user of the first request (without logon ticket). If you use multiple users for calls to the REST interface, you have to ensure that the tickets are only used for the correct user.

If the application calling the REST interface is an application deployed on an SAP NetWeaver AS for Java, for example, a portal application, you may want to reuse the logon ticket already issued for the logged-on user.

The following code snippet shows, how an existing logon ticket of the logged-on user is received from the UME interface and forwarded with the HTTP request:

```
IUser loggedInUser = UMFactory.getAuthenticator().getLoggedInUser();
if (loggedInUser != null){
    String sapLogonTicket=(String)loggedInUser.getTransientAttribute(
        IPrincipal.DEFAULT_NAMESPACE,
        ILoginConstants.SSOTICKET_USER_ATTRIBUTE_PURE);
    Map cookieStore = new HashMap();
    cookieStore.put("MYSAPSS02", sapLogonTicket);
    //enrich HTTP request with cookies as shown in the class HTTPClient
    //method setCookies(...) in the example.zip attached to SAP Note 1665777
}
```

Note

If the REST client application and the SAP NetWeaver Identity Management 7.2 REST interface reside on different servers and you want to use existing logon tickets issued by the server running the REST client application, you need to configure a trust relationship for logon tickets between the two servers. For more information about how to configure this relationship with SAP NetWeaver AS for Java based systems, see *Using Logon Tickets* in the SAP Library that applies to your release.

To enable authentication for systems based on SAP NetWeaver AS for Java 7.1 and higher using logon tickets, you must configure the SAP NetWeaver Identity Management 7.2 REST Interface. The interface texts may vary slightly between releases.

Procedure

1. Log on to SAP NetWeaver Administrator at **http(s)://<server>:<port>/nwa**.
2. Choose ► *Configuration Management* ► *Security* ► *Authentication* .
For AS Java 7.31, choose ► *Configuration Management* ► *Security* ► *Authentication* .
3. Enter **sap.com/tc~idm~rest~ear** in the *Policy Configuration Name* filter and set the *Type* filter to *Web*.
4. Select *sap.com/tc~idm~rest~ear*.
5. Change to edit mode using the *Edit* button.
6. Choose the ticket authentication template.
7. Save the changes using the *Save* button.
8. Choose ► *Home* ► *Operation Management* ► *Systems* ► *Start & Stop* .
For AS Java 7.31, choose ► *Home* ► *Operations* ► *Systems* ► *Start & Stop* .
9. Choose the *Java Applications* tab.
10. In the application list, filter for *tc~idm~rest~ear*.
11. Select the *tc~idm~rest~ear* application.
12. Stop the application using the *Stop* button -> *On All Instances*.
13. Start the application using the *Start* button -> *On All Instances*. Using Logon Ticketsz

Related Information

[Using Logon Tickets](#)

[Single Sign-On With Logon Tickets \[page 830\]](#)

[Authentication and Authorization \[page 829\]](#)

8.11 General Service URIs

Version 2 of the Identity Management REST interface provides two types of metadata documents to describe the service, the Service Document and the Service Metadata Document.

8.11.1 Service Document

The service document is available at the Service Root URI. It returns a list of all available top-level collections. It does not return a list of the available service operations, like `get requests`. These can be found in the Service Metadata Document.

URI

```
(* [GET] *)  
URI = './idmrestapi/v2/service';
```

Request Parameters

None.

Request Example

```
GET /idmrestapi/v2/service HTTPS/1.1
```

Response Example

```
{ "d" : {  
  "EntitySets": [  
    "task collections", "Schema.entrytypes", "Schema.attributes"  
  ]  
} }
```

8.11.2 Service Metadata Document

The Service Metadata Document describes the data model of the Identity Management OData service.

It describes the available entity types, the relationships among them and the available service operations. For more information, see the OData web page. The service metadata is returned as XML format only as defined by the OData specification. The REST API version 2 metadata is built dynamically based on the current Identity Store configuration.

URI

```
(* [GET] *)  
URI    = './idmrestapi/v2/service/$metadata';
```

Request Parameters

None.

Request Example

```
GET /idmrestapi/v2/service/$metadata HTTPS/1.1
```

Related Information

<http://www.odata.org> ➡

8.12 Entry Access URIs

The following URIs allow accessing Identity Store entries, for example, Persons and Roles. You can query for entries, access a certain entry or access and update entry attributes. You can use tasks with the [This task creates a new entry](#) flag set, only for POST (create) operations. Otherwise an error code and message are returned. You can use tasks, for which the flag is not set, for read and modify operations.

Entry Attributes

The entry attributes, for example, MSKEYVALUE, and DISPLAYNAME, which are returned in any entry response, are defined on the entry type. All attributes of the entry type are marked [Allow](#).

Entry type - MX_PERSON

General | Event tasks | Relations | Attributes

Allowed attributes for this entry type

Attribute	Allow	Mandatory	List	Ext. properties	
MSKEYVALUE	☒	☒	☒		
DISPLAYNAME	☒	☒	☒		
MX_FIRSTNAME	☒	☐	☒		
MX_LASTNAME	☒	☐	☒		
DESCRIPTION	☒	☐	☐		
MX_ACADEMIC_TITLE_1	☒	☐	☐		
MX_ACADEMIC_TITLE_2	☒	☐	☐		

However, the attribute values are only set for those attributes contained in the specified task of the respective entry URI. To avoid empty attributes in an entry response, you can use custom query options in search entries or get entry requests.

Attribute Prefixes

Due to the underlying data model and to avoid name conflicts, the attribute names of an entry type are not returned 1:1 in the entry response. Every attribute gets a prefix depending on the type of the attribute.

Attribute Prefixes

Attribute Type	Prefix	Example for Identity Man- agement	Example in REST Interface
Single Value Attribute	SV_	MX_FIRSTNAME	SV_MX_FIRSTNAME

Attribute Type	Prefix	Example for Identity Management	Example in REST Interface
Multi-Value Attribute	MV_	MX_MAIL_ADDITIONAL	MV_MX_MAIL_ADDITIONAL
Entry Reference Attribute	ER_	MXREF_MX_ROLE	ER_MXREF_MX_ROLE
Binary Attributes	BI_	MX_USER_PICTURE	BI_MX_USER_PICTURE

In the system query options of the entry requests, use these prefixes as well when an attribute name is specified, for example, in `$filter` and `$select`.

8.12.1 Access to Security Relevant Attributes

Read and write access to entries require specific handling as compared to accessing task meta information.

The information in this section applies to all read and write accesses for entries.

Returning Encrypted or Hashed Attribute Values

The return values for attributes differ depending on the `hide input` presentation property defined on the task attribute and on whether the connection is secured using SSL or not.

- `hide input` flag is enabled
 - `{HIDDEN}` is returned for non-encrypted, encrypted and hashed attributes.
- `hide input` flag is disabled
 - If the attribute is encrypted and SSL is used, the decrypted value is returned.
 - If the attribute is hashed and SSL is used, the value (hash) is returned.
 - If the attribute is encrypted or hashed but no SSL is used, `{HASHED}` or `{ENCRYPTED}` is returned.

If the attribute does not have a value, `null` is returned.

Encrypted Attributes

To set values for attributes with enabled encryption flag, you must use SSL. Otherwise, the value is not stored and an error with a fitting message is returned to the client. If you use SSL, the JMX layer encrypts the value specified in the HTTP request. The value is then stored in the identity store.

Hashed Attributes

To set values for hashed attributes, you must use SSL. Otherwise, the value is not stored and an error with a fitting message is returned to the client. If you use SSL, the JMX layer encrypts the value specified in the HTTP request. The value is then stored in the identity store.

Setting the Password for a User

To set the password for a user, you must use SSL, and the client should set the value using the `MX_PASSWORD` attribute. The value is hashed and stored for the `MX_PASSWORD` attribute. If the option "Enable password provisioning" is set on the Identity Store, the value is encrypted and stored for the `MX_ENCRYPTED_PASSWORD` attribute. This is the same behavior as in the web enabled tasks.

Note

If the value for the `MX_ENCRYPTED_PASSWORD` attribute is specified, the value is encrypted and stored for the `MX_ENCRYPTED_PASSWORD` attribute but no value for the `MX_PASSWORD` attribute is set.

8.12.2 Support of Binary Attributes

The REST interface supports binary data like pictures. These attributes are of type `File` defined in the `metadata.xml` document. The `File` type is a complex data type, consisting of meta information about the binary attribute value.

The `File` type has a special `SIZE` pseudo property that contains the size of the media object in bytes.

To access the actual binary value of a binary attribute, append `$value` to the URI of the `File` entry. The complete URI is also contained in the property `media_src` of the `File` entry.

8.12.3 Create Entries

To create an entry, you send an HTTPS POST request to the endpoint of the entry's entry type. For example, if you want to create an entry of entry type `MX_PERSON`, you need to send an HTTPS POST request to the `/ET_MX_PERSON` endpoint.

The request body must contain the following:

- The task GUID of the create task for the entry of the corresponding entry type.
- All mandatory attributes in the create task for the entry of the corresponding entry type.
- (Optional) Not mandatory attributes in the create task for the entry of the corresponding entry type.
If the request body does not contain not mandatory attributes, they are not returned in the response body.

Note

Note that the validation of the provided attribute values against the allowed values is done only on the defined in the legal values *Value list*. For more information, see [Attribute Values \(Attribute\) \[page 999\]](#). For validating against other predefined attribute values (for example defined SQL value or Value Help), the **ALLOWED_VALUES** endpoint should be used and validated on client-side before sending the entry creation request. For more information, see [Get Allowed Values of a Specific Task Attribute \[page 922\]](#).

Request

URI: for creating an entry of entry type MX_PERSON: /idmrestapi/v2/service/ET_MX_PERSON

URI: for creating an entry of entry type MX_PRIVILEGE: /idmrestapi/v2/service/ET_MX_PRIVILEGE

URI: for creating an entry of entry type MX_ROLE: /idmrestapi/v2/service/ET_MX_ROLE

HTTPS Method: *POST*

Request Parameters

No request parameters. But a document defining an entry with the attributes has to be passed.

Request Example 1

```
{
  "TASK_GUID" : "65EF6F8B-E0E3-4E11-B092-C82BC6F57376",
  "SV_MSKEYVALUE" : "DupontI",
  "SV_DISPLAYNAME" : "Isabelle Dupont",
  "SV_MX_FIRSTNAME" : "Isabelle",
  "SV_MX_LASTNAME" : "Dupont"
}
```

Request Example 2

This request example contains single value, multi-value and entry reference attributes.

Note

When using the REST API v 2 to create or update an entry with attribute of *Data Type Numeric* and *Presentation SingleLine*, note that the attribute value is saved in the Identity Management database in double format. For example: 1.0

```
{
  "TASK_GUID": "65EF6F8B-E0E3-4E11-B092-C82BC6F57376",
  "SV_MSKEYVALUE": "SmithD",
  "SV_MX_FIRSTNAME": "Denise",
  "SV_MX_LASTNAME": "Smith",
  "SV_DISPLAYNAME": "Denise Smith",
  "ER_MXREF_MX_PRIVILEGE": [
    {
      "REFERENCED_ID": 6,
      "REASON": "New assignment request"
    },
    {
      "REFERENCED_ID": 8,
```

```

        "VALID_TO" : "\Date(1606892760000)\/"
    },
    "MV_MX_USER_PARAMS": [
        {
            "VALUE": "parameter1"
        },
        {
            "VALUE": "parameter2"
        }
    ]
}

```

Response Status and Error Codes

Code	Reason	Description
201	Created	The response body contains the created entry.
400	Bad Request	- The specified task GUID is not a create task. - The request body does not contain task GUID. - The specified entry type name in the URI is wrong. - The request body contains attributes that are not specified in the create task. - The request body does not contain one or more mandatory attributes in the create task.
401	Unauthorized	The user is not authenticated.
403	Forbidden	The user is missing authorizations or the XSRF token is missing.
404	Not Found	No task was found for the specified task GUID.

Response Example

```

{
  "d": {
    "__metadata": {
      "id": "https://.//idmrestapi/v2/service/ET_MX_PERSON(ID=972,TASK_GUID=guid'65ef6f8b-e0e3-4e11-b092-c82bc6f57376')",
      "uri": "https://.//idmrestapi/v2/service/ET_MX_PERSON(ID=972,TASK_GUID=guid'65ef6f8b-e0e3-4e11-b092-c82bc6f57376')",
      "type": "IdmRestModel.ET_MX_PERSON"
    },
    "ID": 972,
    "TASK_GUID": "65ef6f8b-e0e3-4e11-b092-c82bc6f57376",
    "SV_MSKEYVALUE": "DupontI",
    "SV_DISPLAYNAME": "Isabelle Dupont",
    "SV_MX_LASTNAME": "Dupont",
    "SV_MX_FIRSTNAME": "Isabelle"
  }
}

```


8.12.4 Search Entries

Searches for identity store entries of the specified entry type name (MX_ENTRYTYPE).

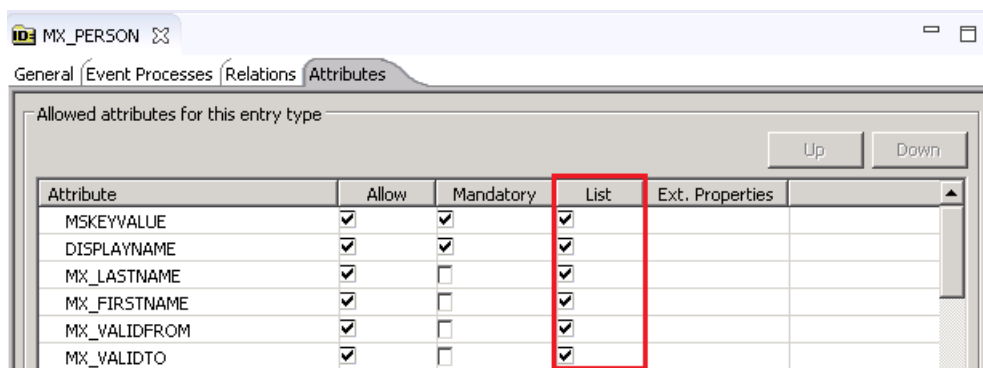
It supports one optional OData parameter (`$filter`) where a search expression with search attributes can be specified. And the custom query option (`filterBasic`) to specify the search filter for the basic search. Also a simple combined search is possible by using both parameters. The search result is a collection of entries.

In the current version 2, the REST interface supports only the “and” operator.

Search Attributes

The attributes that can be used in the search depend on the definition of a search and/or display task for the respective entry type. The determination order is the following:

1. If a search task is defined for the entry type, any attribute contained in this task can be used.
2. If only a display task is defined for the entry type, any attribute contained in this task can be used.
3. If neither a search, nor a display task is set, any attribute that is selected in the [List](#) column of the entry type can be used.



Attribute	Allow	Mandatory	List	Ext. Properties
MSKEYVALUE	☒	☒	☒	
DISPLAYNAME	☒	☒	☒	
MX_LASTNAME	☒	☐	☒	
MX_FIRSTNAME	☒	☐	☒	
MX_VALIDFROM	☒	☐	☒	
MX_VALIDTO	☒	☐	☒	

4. If also no attribute in the [List](#) column is selected, the default search attribute is the MSKEYVALUE.

Search Result

The attribute values of an entry returned in the search result are the attributes selected in the [List](#) column of the corresponding entry type. In addition to those, the MSKEY is returned for every entry as this is a key (as defined in the metadata document).

Note

Every response must have a task GUID, because it is a part of the key of an entry. However, the returned attribute values of the search result are not determined by a task, but by the attributes marked as “List” on the entry type. Therefore, the special task GUID FFFFFFFF -FFFF -FFFF -FFFFFFFFFFFFFF is used in a search response that indicates this behavior.

Maximum Search Results

The maximum amount of entries returned depends on the [Max search result size](#) property, which is set on the [General](#) tab of the Identity Store in the Identity Center. If a search result exceeds this number, the maximum amount of entries is returned in the response. And by default the response will contain a “__next” property with a link that contains the OData query option \$skiptoken with the value `searchLimitExceeded`. This indicates that the search result may not be complete. If a client follows this link, it only gets a 404 - Not Found response, because paging of partial search results is not implemented yet by the Identity Management Rest Service. To deactivate this behavior and to not show the “__next” property in the search result, set the application property `v2.ShowNextLinkWhenSearchLimitIsExceeded` to **false**.

1. In the SAP NetWeaver Administrator, choose ► [Configuration](#) ► [Infrastructure](#) ► [Java System Properties](#) ►.
2. On the [Applications](#) tab page, filter for `tc~idm~rest~ear`.
3. On the [Properties](#) tab page, select `v2.ShowNextLinkWhenSearchLimitIsExceeded`, choose [Modify](#), and enter **false** in the [Enter Custom Value](#) field.

Handling Datetime Format

You can filter entries by validity. The REST API v2 accepts the value of `MX_VALIDFROM` and `MX_VALIDTO` attributes in the following format: **`datetime 'YYYY-MM-DDTHH:MM:SS'`** See Request Example 5.

This request returns all entries matching the given date 'YYYY-MM-DD'.

For more information, see [Attribute DateTime Handling \[page 53\]](#).

URI

```
(* [GET] *)
URI                               = './idmrestapi/v2/service/ET_' EntryTypeName;
EntryTypeName                     = Name of an entry type (as specified in attribute
MX_ENTRYTYPE of each entry) that exists in the Identity Store schema;
```

Request Parameters

ⓘ Note

The URI has to be URL escaped to replace special characters, like the space with “%20”.

Status Search

```
Parameter      = ['?' BasicFilter]
BasicFilter    = 'filterBasic=' SearchValue;
SearchValue    = Edm.String; (* * is wildcard *)
```

With this filter parameter an OR search is performed for all search attributes (see above).

Advanced Search

```
Parameter      = '?' Filter
Filter         = '$filter=' SearchAttributes;
SearchAttributes = KeyValuePair (Junction KeyValuePair)*;
Junction       = ' and '; (* case sensitive! *)
KeyValuePair    = FilterAttribute Name Comparator AttributeValue;
FilterAttrName  = AttributePrefix AttributeName
AttributePrefix = 'SV_' | 'MV_' | 'ER_' (* 'SV_' for single-value
attribute, 'MV_' for multi-value attributes, or 'ER_' for entry reference
attributes *)
Comparator      = Equal
Equal           = ' eq '; (* Operator eq is case sensitive! *)
AttributeName   = Edm.String; (* see explanation below *)
AttributeValue  = "" Edm.String ""; (* * is wildcard *)
```

Note

Searching by entry reference attributes is supported in [Advanced](#) search only.

Search Operators

The REST interface supports in the first version only the “and” operator.

Allowed Attribute Names

- The attribute names allowed in the search filter are the ones from the search attributes (see above).
- If a specified attribute name is not contained in the search attributes, the request is invalid. This results in a 400 - Bad Request response.

Attribute with Value Help

- For attributes that have a value help, the value key has to be specified as search value. A search for the value text is not supported.

Combined Search

```
Parameter      = '?' BasicFilter '&' Filter
```

- The combined search logically combines the results for both searches with an “AND” operator.
- Only one search attribute is supported for the advanced search filter.

Note

Combined search supports only single-value and multi-value non-reference attributes for the **Filter** parameter.

Request Example 1

Basic search for all entries of entry type MX_PERSON.

```
GET /idmrestapi/v2/service/ET_MX_PERSON
```

Request Example 2

Advanced search for entries of entry type MX_PERSON with the first name "Mirco*" and the last name "Schuster*".

```
GET /idmrestapi/v2/service/ET_MX_PERSON?$filter=SV_MX_FIRSTNAME eq 'Mirco*' and  
SV_MX_LASTNAME eq 'Schuster*'
```

Request Example 3

Basic search for entries of entry type MX_PERSON with the value "Mirco" in one of the search attributes.

```
GET /idmrestapi/v2/service/ET_MX_PERSON?filterBasic=Mirco
```

Request Example 4

Combined search for entries of entry type MX_PERSON with the value "Mirco" in one of the search attributes and the department "Sales".

```
GET /idmrestapi/v2/service/ET_MX_PERSON?  
filterBasic=Mirco&$filter=SV_MX_DEPARTMENT eq 'Sales'
```

Combined search for entries of entry type MX_PERSON with the value "Mirco" in one of the search attributes and the additional email "mirco@mail.com".

```
GET /idmrestapi/v2/service/ET_MX_PERSON?  
filterBasic=Mirco&$filter=MV_MX_MAIL_ADDITIONAL/VALUE eq 'mirco@mail.com'
```

Request Example 5

Search for entries of entry type MX_PERSON with validity from "datetime'2018-10-01T02:00:00".

```
GET /idmrestapi/v2/service/ET_MX_PERSON?$filter=SV_MX_VALIDTFROM eq
datetime'2018-10-01T02:00:00'
```

Request Example 6

When filtering for multi-value attributes, the following syntax should be used:

```
MV_<attribute>/VALUE eq '<search value>'
```

You can also combine filtering for multi-value attributes with \$expand and \$select options.

```
GET /idmrestapi/v2/service/ET_MX_PERSON?$filter=MV_MX_MAIL_ADDITIONAL/VALUE
eq '*abcd*' & $select=MV_MX_MAIL_ADDITIONAL,
SV_DISPLAYNAME&$expand=MV_MX_MAIL_ADDITIONAL
```

Request Example 7

When searching for entries by entry reference attribute, this attribute should be in the [List](#) column of the corresponding entry type.

You can filter the referenced attributes via REFERENCED_ID(MSKEY), REFERENCED_MSKEYVALUE(MSKEYVALUE) or REFERENCED_DISPLAY_NAME (DISPLAYNAME).

You can find more examples below:

```
GET /idmrestapi/v2/service/ET_MX_PERSON?$filter=ER_MX_MANAGER/REFERENCED_ID eq
<MSKEY>
```

```
GET /idmrestapi/v2/service/ET_MX_PERSON?$filter=ER_MXREF_MX_PRIVILEGE/
REFERENCED_MSKEYVALUE eq '<MSKEYVALUE>'
```

```
GET /idmrestapi/v2/service/ET_MX_PERSON?$filter=ER_<CUSTOM_REFERENCE_ATTRIBUTE>/
REFERENCED_DISPLAY_NAME eq '<DISPLAYNAME>'
```

As a response you will receive all entries, with their attributes and values, which have the referenced attribute assigned.

Response Types

```
Content-Type: application/json | application/atom+xml
```

HTTPS Response Codes

- 200 OK
Response Body: Collection of entities with the type of the requested entry type (MX_ENTRYTYPE).
- 400 Bad Request
 - If the `$filter` parameter does not comply with the rules defined in section *Parameters* or if the attribute name is not an attribute of the corresponding entry type.
 - If the way the `$filter` and `filterBasic` parameters are used together is not supported.
- 403 Forbidden
Logged-in user is not allowed to search for entries.
- 404 Not Found
If the specified entry type name does not exist.

Response Example 1

```
{
  "d": {
    "results": [
      {
        "__metadata": {
          "uri": "/idmrestapi/v2/service/ET_MX_PERSON(ID=4321,TASK_GUID=guid'FFFFFFFF-FFFF-FFFF-FFFFFFFFFFFF')",
          "type": "MX_PERSON",
        },
        "ID": 4321,
        "SV_MSKEYVALUE": "mircoschuster1",
        "SV_DISPLAYNAME": "Mirco Schuster",
        "SV_MX_FIRSTNAME": "Mirco",
        "SV_MX_LASTNAME": "Schuster",
      },
      {
        "__metadata": {
          "uri": "/idmrestapi/v2/service/ET_MX_PERSON(ID=4322,TASK_GUID=guid'FFFFFFFF-FFFF-FFFF-FFFFFFFFFFFF')",
          "type": "MX_PERSON",
        },
        "ID": 4322,
        "SV_MSKEYVALUE": "moschuessler1",
        "SV_DISPLAYNAME": "Mo Schüssler",
        "SV_MX_FIRSTNAME": "Mo",
        "SV_MX_LASTNAME": "Schüssler",
      }
    ]
  }
}
```

Response Example 2

```
{
  "d": {
    "results": [
      {
        "__metadata": {
          "id": "/idmrestapi/v2/service/ET_MX_PERSON(ID=143,TASK_GUID=guid'ffffffff-ffff-ffff-ffff-ffffffffffff')",

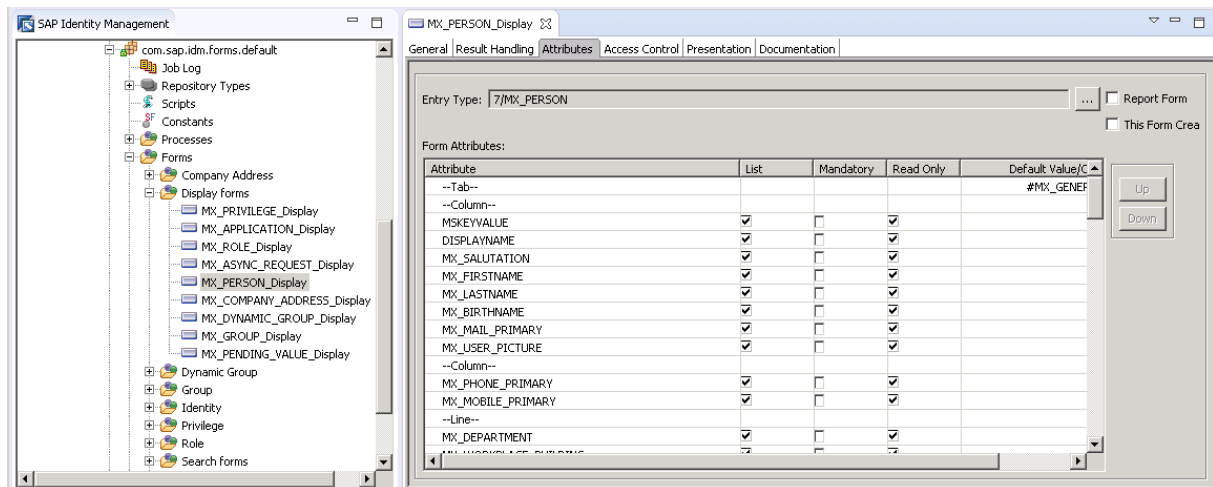
```

```
"uri": "/idmrestapi/v2/service/  
ET_MX_PERSON(ID=143,TASK_GUID=guid'ffffffff-ffff-ffff-ffff-fffffffffffffff')",  
  "type": "IdMRestModel.ET_MX_PERSON"  
},  
  "SV_DISPLAYNAME": "Mirco",  
  "MV_MX_MAIL_ADDITIONAL": {  
    "results": [  
      {  
        "__metadata": {  
          "id":  
"/idmrestapi/v2/service/MV_MX_MAIL_ADDITIONAL(ID=28,TASK_GUID=guid'ffffffff-ffff-  
ffff-ffff-fffffffffffffff',VALUE='abcd1234%40sap.com')",  
          "uri":  
"/idmrestapi/v2/service/MV_MX_MAIL_ADDITIONAL(ID=28,TASK_GUID=guid'ffffffff-ffff-  
ffff-ffff-fffffffffffffff',VALUE='abcd1234%40sap.com')",  
          "type": "IdMRestModel.MultiValueString"  
        },  
        "ID": 28,  
        "TASK_GUID": "ffffffff-ffff-ffff-ffff-fffffffffffffff",  
        "VALUE": "abcd1234@sap.com",  
        "OPERATION": "0"  
      },  
      {  
        "__metadata": {  
          "id":  
"/idmrestapi/v2/service/MV_MX_MAIL_ADDITIONAL(ID=28,TASK_GUID=guid'ffffffff-ffff-  
ffff-ffff-fffffffffffffff',VALUE='abcd%40sap.com')",  
          "uri":  
"/idmrestapi/v2/service/MV_MX_MAIL_ADDITIONAL(ID=28,TASK_GUID=guid'ffffffff-ffff-  
ffff-ffff-fffffffffffffff',VALUE='abcd%40sap.com')",  
          "type": "IdMRestModel.MultiValueString"  
        },  
        "ID": 28,  
        "TASK_GUID": "ffffffff-ffff-ffff-ffff-fffffffffffffff",  
        "VALUE": "abcd@sap.com",  
        "OPERATION": "0"  
      }  
    ]  
  }  
}  
]  
}  
}  
}
```

8.12.5 Get Specific Entry

Returns a specific entry for a given MSKEY containing a set of its attributes that are defined by the passed task and are set for the entry.

The TaskGUID must originate from a UI task and the currently logged-in user must be able to access this task.



The type of an attribute depends on whether it is a single-value, a multi-value or an entry reference attribute. For more information, see *Get Attribute Values of an Entry*.

For the single-value attributes that do not contain a value, the `Null` value is returned. If you don't want the `Null` value to be returned in the response body, set the `v2ReturnNullValuesInResponse` Java system property of the deployed `tc~idm~rest~ear` application on SAP NetWeaver AS Java to `false`. The default value is `true`. For more information, see [Configuring the Java System Properties](#).

URI

```
(* [GET] *)
URI = './idmrestapi/v2/service/ET_' EntryType '(ID=' Mskey
',TASK_GUID=' TaskGUID ')';
EntryType = Edm.String;
Mskey = Edm.Int32;
TaskGUID = Edm.Guid;
```

Note

The `IdStoreID` is not required here, because the `MSKEY` is unique across all Identity Stores.

Request Parameters

Supported Query Options:

Name	Description
\$expand	To inline deferred multi-value or entry reference attributes.

Name	Description
\$select	To only return a subset of the attributes that are exposed by the specified task.

Request Example 1

```
GET /idmrestapi/v2/service/ET_MX_PERSON(ID=7894,TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE') HTTPS/1.1
```

Request Example 2

```
GET /idmrestapi/v2/service/ET_MX_PERSON(ID=7894,TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE')?$expand=ER_MXREF_MX_ROLE HTTPS/1.1
```

Request Example 3

```
GET /idmrestapi/v2/service/ET_MX_PERSON(ID=7894,TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE')?$expand=MV_MX_PHONE_ADDITIONAL HTTPS/1.1
```

Response Types

```
Content-Type: application/json
```

HTTPS Status Codes:

- 200 OK
Response Body: Entry
- 400 Bad Request
 - Task is a create task.
- 403 Forbidden
 - Logged-in user is not allowed to access the task.
 - Task is a self-service task and the MSKEY is not from the logged-in user.
- 404 Not Found
If the specified entry does not exist .

Response Example 1

```
{ "d" : {
  "results":{
    "__metadata":{
      "uri":"/idmrestapi/v2/service/
ET_MX_PERSON(ID=7894,TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE')",
      "type" : "MX_PERSON"
    },
    "ID" : 7894,
    "TASK_GUID": "35B264BF-A75A-447D-A4EA-7894725245CE",
    "SV_MSKEYVALUE" : "schuessler1",
    "SV_MX_LASTNAME" : "Schüssler",
    "SV_MX_FIRSTNAME" : "Mo",
    "SV_MX_MAIL_PRIMARY" : "mo.schuessler@example.com",
    "SV_DISPLAYNAME" : "Mo Schüssler",
    "MV_MX_PHONE_ADDITIONAL" : {
      "__deferred" : {
        "uri : "/idmrestapi/v2/service/ET_MX_PERSON(ID=7894,TASK_GUID=guid'35B264BF-
A75A-447D-A4EA-7894725245CE')/MV_MX_PHONE_ADDITIONAL"
      }
    },
    "ER_MXREF_MX_ROLE" : {
      "__deferred" : {
        "uri : "/idmrestapi/v2/service/ET_MX_PERSON(ID=7894,TASK_GUID=guid'35B264BF-
A75A-447D-A4EA-7894725245CE')/ER_MXREF_MX_ROLE"
      }
    }
  }
} }
```

ⓘ Note

For more information about attributes that contain binary data (for example, `BI_MX_USER_PICTURE`, which can contain pictures with content types like image or jpeg), see *Get Attribute Values of an Entry*.

Response Example 2

```
{ "d" : {
  "results": {
    "__metadata" : {
      "uri" : "/idmrestapi/v2/service/
ET_MX_PERSON(ID=7894,TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE')",
      "type" : "MX_PERSON"
    },
    "ID" : 7894,
    "TASK_GUID" : "35B264BF-A75A-447D-A4EA-7894725245CE",
    "SV_MSKEYVALUE" : "schuessler1",
    "SV_MX_LASTNAME" : "Mo",
    "SV_MX_FIRSTNAME" : "Schuessler",
    "SV_MX_MAIL_PRIMARY" : "mo.schuessler@example.com",
    "SV_DISPLAYNAME" : "Mo Schüssler",
    "MV_MX_PHONE_ADDITIONAL" : {
      "__deferred" : {
```

```

"uri" : "/idmrestapi/v2/service/ET_MX_PERSON(ID=7894,TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE')/MV_MX_PHONE_ADDITIONAL"
    }
  },
  "ER_MXREF_MX_ROLE" : [ {
    "__metadata" : {
      "uri" : "/idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=1737,TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE')",
      "type" : "EntryReference"
    },
    "TASK_GUID" : "35B264BF-A75A-447D-A4EA-7894725245CE",
    "LINK_ID" : 1737,
    "REFERENCED_ID" : 41203,
    "REFERENCED_DISPLAY_NAME" : "role 3",
    "VALID_FROM" : "\/Date(1360454400000)\/",
    "VALID_TO" : "\/Date(1362095999000)\/",
    "REASON" : "undefined - test",
    "STATE" : 4,
    "STATE_TEXT" : "Pending",
    "DIRECTLY_ASSIGNED" : true,
    "INHERITED" : false,
    "OPERATION" : null,
    "CONTEXT" : null,
    "REFERENCED_ENTRY": {
      "__deferred": {
        "uri": "/"
        idmrestapi/v2/service/ER_MXREF_MX_PRIVILEGE(LINK_ID=1737,TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE')/REFERENCED_ENTRY"
      }
    },
    "CONTEXT_ENTRY": {
      "__deferred": {
        "uri": "/"
        idmrestapi/v2/service/ER_MXREF_MX_PRIVILEGE(LINK_ID=1737,TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE')/CONTEXT_ENTRY"
      }
    }
  } ],
  {
    "__metadata" : {
      "uri" : "/idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=1738,TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE')",
      "type" : "EntryReference",
    },
    "TASK_GUID" : "35B264BF-A75A-447D-A4EA-7894725245CE",
    "LINK_ID" : 1738,
    "REFERENCED_ID" : 41204,
    "REFERENCED_DISPLAY_NAME" : "role 4",
    "VALID_FROM" : null,
    "VALID_TO" : null,
    "REASON" : null,
    "STATE" : 3,
    "STATE_TEXT" : "Failed",
    "DIRECTLY_ASSIGNED" : true,
    "INHERITED" : false,
    "OPERATION" : null,
    "CONTEXT" : null,
    "REFERENCED_ENTRY": {
      "__deferred": {
        "uri": "/"
        idmrestapi/v2/service/ER_MXREF_MX_PRIVILEGE(LINK_ID=1738,TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE')/REFERENCED_ENTRY"
      }
    },
    "CONTEXT_ENTRY": {
      "__deferred": {
        "uri": "/"
        idmrestapi/v2/service/ER_MXREF_MX_PRIVILEGE(LINK_ID=1738,TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE')/CONTEXT_ENTRY"
      }
    }
  }
]

```



```

        "OPERATION" : null
      }, {
        "__metadata" : {
          "uri" : "/idmrestapi/v2/service/MV_MX_PHONE_ADDITIONAL(ID=7894,
TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE',VALUE='+496123199999999'),
          "type" : "MultiValueString"
        }
      }, {
        "VALUE" : "+496123199999999",
        "OPERATION" : null
      } ],
      "ER_MXREF_MX_ROLE" : {
        "__deferred" : {
          "uri" : "/idmrestapi/v2/service/ET_MX_PERSON(ID=7894,TASK_GUID=guid'35B264BF-
A75A-447D-A4EA-7894725245CE')/ER_MXREF_MX_ROLE"
        }
      }
    }
  }
} }

```

Related Information

[Get Attribute Values of an Entry \[page 870\]](#)

8.12.6 Update an Entry

Updates an entry specified by its MSKEY with any of the attributes defined by the passed TaskGUID.

Prerequisites

- The task is a UI task.
- The logged-in user is authorized to access this task.
- The attributes being written are not read-only.

Note

Note that the validation of the provided attribute values against the allowed values is done only on the defined in the legal values [Value list](#). For more information, see [Attribute Values \(Attribute\) \[page 999\]](#). For validating against other predefined attribute values (for example defined SQL value or Value Help), the **ALLOWED_VALUES** endpoint should be used and validated on client-side before sending the entry update request. For more information, see [Get Allowed Values of a Specific Task Attribute \[page 922\]](#).

MERGE Method via Method Tunneling through POST

The MERGE operation must be requested using method tunneling through POST including the header X-HTTP-Method with the value MERGE. If the update is called via the POST HTTPS method with MERGE method

tunneling, only the attributes that should be updated have to be passed in the request document. If an attribute that is exposed by the specified task is not passed, the update does not change it.

Update Attributes with Value Help

For attributes with value help, you must specify the value key as the new value. If you specify the value text, an error occurs.

Update Multi-value and Entry Reference Attributes

You can add and update single values for a multi-value attribute or an entry reference using a merge entry request. Unspecified values or entry references remain unchanged.

You can specify the **OPERATION** property for single changes in an update request. With this operation you can combine different operations in one request without a **BATCH** request, as the **BATCH** request is not fully supported by the odata4j library. With the **OPERATION** property you can add, delete, and modify multi-value or entry reference attributes with a single update request.

Possible Operations for Entry References or Multi-Value Attributes

OPERATION Property Values

Operation Value	Comment
ADD	Entry Reference: An entry reference can be created by specifying the REFERENCED_ID property. Multi-Value: Indicating the VALUE property, you can add a value to the multi-value attribute.
MOD	Entry Reference: An entry reference identified by the LINK_ID can be modified. Multi-Value: This operation is not supported for multi-value attributes.
REPLACE	When adding an entry reference or multi-value with the REPLACE operation, all current entry references or multi-values (of the specific attribute) are removed and replaced with the one specified. If more entry reference or multi-values are added to the same attribute and at least one has the REPLACE operation, all others also must have the REPLACE operation.

Operation Value	Comment
DEL	Entry Reference: An entry reference identified by the LINK_ID property can be deleted. Multi-Value: A value of the multi-value attribute identified by the VALUE property can be deleted.
DEL_ALL	If this operation was specified, all entry references or multi-values (of the specific attribute) are deleted. When using the DEL_ALL operation, no other entry references or multi-values (of the specific attribute) can be added in the same HTTPS request. If you want to do so, use the REPLACE operator instead.
<null> or OPERATION property not set	Entry References: If no operation was specified the following applies: LINK_ID specified: A modification of the entry reference is executed. If no LINK_ID was specified the REFERENCED_ID property is mandatory; an ADD operation is executed. Multi-value: If you do not indicate any operation, the operation ADD (default) is used.

Handling of Entry Reference Attributes

If the attribute is an entry reference, for example, **MXREF_MX_PRIVILEGE**, the specification of only the attribute name results in an error. When creating an assignment (entry reference), you can specify the following properties.

Assignment Properties

Property	Comment
LINK_ID	Must be null or do not specify this property.
REFERENCED_ID	Mandatory
VALID_FROM	Optional
VALID_TO	Optional
REASON	Optional
CONTEXT	Complex type CONTEXT is optional.
CONTEXT_ID	Property ID of CONTEXT is optional.
OPERATION	Optional or ADD value.

When modifying an assignment (entry reference), you can specify the following properties. If you do not specify the `VALID_FROM` or `VALID_TO` property, it is stored as `null` in the database.

Assignment Properties

Property	Comment
LINK_ID	Mandatory
REFERENCED_ID	Optional
VALID_FROM	Optional
VALID_TO	Optional
REASON	Optional
OPERATION	Optional or <code>MOD</code> value.

When deleting an assignment (entry reference), you can specify the following properties.

Assignment Properties

Property	Comment
LINK_ID	Mandatory
REASON	Optional
OPERATION	Mandatory, value must equal <code>DEL</code> .

Handling of Multi-Value Attributes

Changing a multi-value attribute is similar to changing an entry reference attribute. You can only change a single value using a collection access as shown in *Request Example 4*. When creating a value for a multi-value attribute, you can specify the following properties.

Multi-Value Attribute Properties

Property	Comment
Value	Mandatory
OPERATION	Optional or value <code>ADD</code> .

When deleting a value of a multi-value attribute, you must specify the following properties.

Property	Comment
Value	Mandatory
OPERATION	Mandatory, value must equal DEL.

Delete an Entry

In contrast to the UI tasks that create an entry and can be identified by the [This task creates a new entry](#) option, there is no equivalent for tasks that delete an entry. These tasks are normal update tasks, which have a dependent task together with a link to a job that will then delete the entry. Therefore the URI format for updating and deleting an entry is the same.

To delete an entry, you must send a POST request with an empty body. If the request body contains read-only attributes, such as MSKEYVALUE, you will get an error that read-only attributes cannot be modified. If the request body contains modifiable attributes, the entry will be updated first, and then deleted. See Request Example 3.

URI

```
(* [POST with Merge-Method tunneling] *)
URI          = './idmrestapi/v2/service/ET_'EntryType'(ID=' Mskey
',TASK_GUID=' TaskGUID ');
EntryType    = Edm.String;
Mskey        = Edm.Int32;
TaskGUID     = Edm.Guid;
```

Note

The IdStoreID is not required here, because the MSKEY is unique across all Identity Stores.

Request Parameters

No request parameters. But a document defining an entry with the attributes has to be passed.

Request Example 1

```
POST /idmrestapi/v2/service/
ET_MX_PERSON(ID=7894,TASK_GUID=guid'D092EA23-9E5D-4FC7-9467-1E1D0BBE3629')/
HTTPS/1.1
```

X-HTTP-METHOD:MERGE

```
{
  "SV_MX_LASTNAME" : "Benz",
  "SV_MX_FIRSTNAME" : "Marie",
  "SV_MX_MAIL_PRIMARY" : "marie.benz@example.com",
  "SV_DISPLAYNAME" : "Marie Benz",
  "SV_MX_SALUTATION" : "Ms.",
  "ER_MXREF_MX_ROLE" : [ {
    "LINK_ID" : 1737,
    "REASON" : "Extend validity date for existing assignment",
    "VALID_FROM" : "\/Date(1360454400000)\/",
    "VALID_TO" : "\/Date(1372095999000)\/",
  }, {
    "REFERENCED_ID" : 41204,
    "REASON" : "New assignment request with validity",
    "VALID_FROM" : "\/Date(1360454400000)\/",
    "VALID_TO" : "\/Date(1362095999000)\/",
  }, {
    "REFERENCED_ID" : 41205,
    "REASON" : "New assignment request with context only",
    "CONTEXT" : {
      "CONTEXT_ID" : 78
    }
  }, {
    "LINK_ID" : 1738,
    "REASON" : "Delete an assignment via DEL operator",
    "OPERATION" : "DEL"
  } ],
  "MV_MX_PHONE_ADDITIONAL" : [{
    "VALUE" : "+012345689"
  }, {
    "VALUE" : "+987654321"
  } ],
  "MV_MX_FAX_ADDITIONAL" : {
    "VALUE" : "+12121212"
  }
}
```

Note

To change the `VALID_TO` value, you must also indicate the `VALID_FROM` value. If you do not want to change `VALID_FROM` value, indicate the current `VALID_FROM` value.

Request Example 2

```
POST /idmrestapi/v2/service/
ET_MX_PERSON(ID=7894,TASK_GUID=guid'D092EA23-9E5D-4FC7-9467-1E1D0BBE3629')/
HTTPS/1.1
X-HTTP-METHOD:MERGE
```

```
{
  "ER_MXREF_MX_ROLE" : {
    "OPERATION" : "DEL_ALL"
  },
  "MV_MX_PHONE_ADDITIONAL" : [{
    "VALUE" : "+092345689",
    "OPERATION": "REPLACE"
  }, {
```

```
        "VALUE" : "+9187654321" ,  
        "OPERATION": "REPLACE"  
    }  
}
```

Request Example 3

```
POST /idmrestapi/v2/service/  
ET_MX_PERSON(ID=511,TASK_GUID=guid'34A1E7BC-7A50-48E8-A508-36BDD4A02F01')/  
HTTPS/1.1
```

```
{}
```

Response Types

None.

HTTPS Response Codes

- 204 – No Content
- 400 Bad Request
A mandatory attribute of the entry is missing or a read-only attribute of the entry is specified.
- 401 – Unauthorized
The user is not authenticated.
- 403 – Forbidden
The user is missing authorizations or the XSRF token is missing.
- 404 – Not found
The specified task or entry was not found. The response body contains information about the not found entity.

Response Example

```
HTTPS/1.1 204 No Content
```

Related Information

<http://www.odata.org/developers/protocols/operations#UpdatingEntries> ➤

[Add Attribute Values to an Entry \[page 878\]](#)

[Update Attribute Values of an Entry \[page 882\]](#)

8.12.7 Get Attribute Values of an Entry

Returns the value(s) for a single attribute of an entry, specified by the entry's MSKEY, the GUID of the task containing the attribute, and the attribute name.

The specified attribute has to be exposed by the task, passed via the TaskGUID. The task has to be a UI task and the currently logged-in user must be authorized to access this task.

For accessing the “raw” value of a primitive property, `$value` can be used as described in the OData specification. See also Request Example 2. In Identity Management “Raw” value means the pure string representation of the attribute value, as stored in the Identity Center database (for example, the content of the `aValue/aLong` field in `idmv_value_*` view) value of an attribute.

Multi-value and Entry Reference Attributes

To allow the manipulation (add, modify, delete) of single attribute values and entry references, the access to all multi-value and entry reference attributes is done via an access to an Entity Set (collection of entries).

This means, the GET request for an attribute value on these types of attributes returns a collection of either a value or an entry reference. To retrieve a specific value or reference, specify the key of the reference. For multi-value attributes the key is the value itself; for entry references it is the link ID.

URI

```
( * [GET] *)
```

- Single-value Attribute

```
URI = './idmrestapi/v2/service/ET_' EntryType '(ID=' Mskey  
,TASK_GUID=' TaskGUID ')/' AttributeName ['/ $value' ];
```

- Multi-value Attribute:

```
Attribute is a collection of entries of type "MultiValueString",  
"MultiValueDouble", or "MultiValueDateTime"  
- Get one value:  
URI = './idmrestapi/v2/service/MV_' MultiValue '(ID=' Mskey  
,TASK_GUID=' TaskGUID ')/' AttributeName '(' Value ' ' ;  
- Get all values:  
URI = './idmrestapi/v2/service/MV_' MultiValue '(ID=' Mskey  
,TASK_GUID=' TaskGUID ')/' AttributeName;
```

- Entry Reference Attribute:

```
Attribute is a collection of entries of type "EntryReference"
```

```

- Get one reference:
URI      = './idmrestapi/v2/service/ER_' EntryReference '(ID=' Mskey
',TASK_GUID=' TaskGUID ')/' AttributeName '(' LinkId ')' ;
- Get all (direct) references:
URI      = './idmrestapi/v2/service/ER_' EntryReference '(ID=' Mskey
',TASK_GUID=' TaskGUID ')/' AttributeName;
EntryType      = Edm.String;
MultiValue     = Edm.String;
EntryReference  = Edm.String;
TaskGUID       = Edm.Guid;
Mskey          = Edm.Int32;
AttributeName   = Edm.String;
Value          = Edm.String;
LinkId         = Edm.Int32;

```

Request Parameters

Only the URI for getting all references allows request parameters to specify which assignments should be returned, for example, current, future, failed or only assignments for a certain context (see Request Example 2).

By default only the directly assigned references are returned.

```

Parameter      = [ '?' Filter ] [ '&' ShowIndirect ]
Filter          = '$filter=' SearchAttributes;
ShowIndirect    = 'showIndirect';           // also shows indirectly
assigned references
SearchAttributes = KeyValuePair (Junction KeyValuePair)*;
Junction        = 'and';
KeyValuePair     = PropertyName Comperator PropertyValue;
Comperator      = Equal;
Equal           = 'eq';
PropertyValue    = literalExpression;
literalExpression = stringLiteral / dateTimeLiteral / int16Literal /
booleanLiteral / nullLiteral

```

For the literalExpression, see the *Abstract Type System* section on the OData web page.

Name	Description
showReferencedEntryDetails	Retrieves the attributes defined on the Display form of an entry reference attribute, such as ER_MX_MANAGER, regardless of whether access limitations are configured on entry type level (MX_PERSON). See the request example below. When searching for a user with ER_MX_MANAGER defined on this user, you get the user's manager details in the response. Displaying the manager's details will only be denied, if Use access limitations for display task option is selected on the entry type. This will result in a 403 - Forbidden response.

```
GET /idmrestapi/v2/service/ET_MX_PERSON(ID=35,TASK_GUID=guid'd9aa384a-c5d2-4008-
adb7-6bcd63ddae72')/ER_MX_MANAGER?showReferencedEntryDetails=26
```

Request Example 1

Get all additional phone numbers of the user with ID=7894.

```
GET /idmrestapi/v2/service/ET_MX_PERSON(ID=7894,TASK_GUID=guid'35B264BF-
A75A-447D-A4EA-7894725245CE')/MV_MX_PHONE_ADDITIONAL HTTPS/1.1
```

Request Example 2

```
GET /idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=1755,TASK_GUID=guid'35B264BF-
A75A-447D-A4EA-7894725245CE') HTTPS/1.1
```

Request Example 3

```
GET /idmrestapi/v2/service/ET_MX_PERSON(ID=7894,TASK_GUID=guid'35B264BF-
A75A-447D-A4EA-7894725245CE')/BI_MX_USER_PICTURE/$value
```

Request Example 4

```
GET /idmrestapi/v2/service/BI_MX_USER_PICTURE(ID=7894,TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE')/$value
```

Request Example 5

Get direct and indirect assignments of the user with ID=7894.

```
GET /idmrestapi/v2/service/ET_MX_PERSON(ID=7894,TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE')/ER_MX_ASSIGNMENT?showIndirect
```

Response Types

- Content-Type: application/json | application/atom+xml
- The type of the attribute when using \$value (for example, text/plain for Edm.String).

HTTPS Response Codes

- 200 - OK
- 401 - Unauthorized
If the user is not authenticated.
- 403 - Forbidden
If the user is not authorized to access this attribute.
- 404 - Not Found
- If the requested attribute does not exist in the schema of the entry.

Response Example 1

```
{ "d" : {  
  "results" : [ {  
    "__metadata" : {  
      "uri" : "/idmrestapi/v2/service/MV_MX_PHONE_ADDITIONAL(ID=7894,  
TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE',VALUE='+496123123456789')",  
      "type" : "MultiValueString"  
    }  
    "VALUE" : "+496123123456789"  
    "OPERATION" : null  
  }, {  
    "__metadata" : {
```

```

        "uri" : "/idmrestapi/v2/service/MV_MX_PHONE_ADDITIONAL(ID=7894,
TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE', VALUE='+496123199999999'),
        "type" : "MultiValueString"
    }
    "VALUE" : "+496123199999999"
    "OPERATION" : null
} ]
} }

```

Response Example 2

```

{
  "d": {
    "__metadata": {
      "id": "https://./idmrestapi/v2/service/
ER_MXREF_MX_ROLE(LINK_ID=43,TASK_GUID=guid'102e1d96-1813-4a58-9880-9572056f0fd1')
",
      "uri": "https://./idmrestapi/v2/service/
ER_MXREF_MX_ROLE(LINK_ID=43,TASK_GUID=guid'102e1d96-1813-4a58-9880-9572056f0fd1')
",
      "type": "IdMRestModel.ER_MXREF_MX_ROLE.ET_MX_ROLE.ETEntryReference"
    },
    "TASK_GUID": "102e1d96-1813-4a58-9880-9572056f0fd1",
    "LINK_ID": 43,
    "REFERENCED_ID": 26,
    "REFERENCED_MSKEYVALUE": "AdminRole",
    "REFERENCED_DISPLAY_NAME": "Admin Role",
    "REFERENCED_DESCRIPTION": null,
    "REFERENCED_ENTRYTYPE": "MX_ROLE",
    "VALID_FROM": "/Date(1515542400000)/",
    "VALID_TO": "/Date(1548979199000)/",
    "REASON": "Administrator in Project X",
    "STATE": 1,
    "STATE_TEXT": "OK",
    "DIRECTLY_ASSIGNED": true,
    "INHERITED": false,
    "OPERATION": "0",
    "CONTEXT": {
      "__metadata": {
        "type": "IdMRestModel.CONTEXT"
      },
      "CONTEXT_ID": 25,
      "CONTEXT_DISPLAY_NAME": "Project X",
      "CONTEXT_ENTRYTYPE_DISPLAY_NAME": "Project"
    },
    "DIRECT_VALID_FROM": "/Date(1515542400000)/",
    "DIRECT_VALID_TO": "/Date(1548979199000)/",
    "ASSIGNED_AT": null,
    "ASSIGNED_BY": null,
    "STATUS": 0,
    "REFERENCED_ENTRY": {
      "__deferred": {
        "uri": "https://./idmrestapi/v2/service/
ER_MXREF_MX_ROLE(LINK_ID=43,TASK_GUID=guid'102e1d96-1813-4a58-9880-9572056f0fd1')
/REFERENCED_ENTRY"
      }
    },
    "CONTEXT_ENTRY": {
      "__deferred": {
        "uri": "https://./idmrestapi/v2/service/
ER_MXREF_MX_ROLE(LINK_ID=43,TASK_GUID=guid'102e1d96-1813-4a58-9880-9572056f0fd1')
/CONTEXT_ENTRY"
      }
    }
  }
}

```



```

    },
    "HIERARCHY": {
      "__deferred": {
        "uri": "https://./idmrestapi/v2/service/
ER_MXREF_MX_ROLE(LINK_ID=43,TASK_GUID=guid'102e1d96-1813-4a58-9880-9572056f0fd1')
/HIERARCHY"
      }
    },
    "ASSIGNMENT_HISTORY": {
      "__deferred": {
        "uri": "https://./idmrestapi/v2/service/
ER_MXREF_MX_ROLE(LINK_ID=43,TASK_GUID=guid'102e1d96-1813-4a58-9880-9572056f0fd1')
/ASSIGNMENT_HISTORY"
      }
    }
  }
}

```

Response Example 5

```

{
  "d": {
    "results": [
      {
        "__metadata": {
          "id": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=62,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')",
          "uri": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=62,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')",
          "type": "IdMRestModel.ER_MX_ASSIGNMENT"
        },
        "TASK_GUID": "a56db5fe-da5a-4109-8518-054b7e0fa454",
        "LINK_ID": 62,
        "REFERENCED_ID": 31,
        "REFERENCED_MSKEYVALUE": "Java Developer Role",
        "REFERENCED_DISPLAY_NAME": "Java Developer Role",
        "REFERENCED_DESCRIPTION": null,
        "REFERENCED_ENTRYTYPE": "MX_ROLE",
        "VALID_FROM": "/Date(1515715200000)/",
        "VALID_TO": "/Date(1549065599000)/",
        "REASON": null,
        "STATE": 1,
        "STATE_TEXT": "OK",
        "DIRECTLY_ASSIGNED": true,
        "INHERITED": false,
        "OPERATION": "0",
        "CONTEXT": {
          "__metadata": {
            "type": "IdMRestModel.CONTEXT"
          },
          "CONTEXT_ID": null,
          "CONTEXT_DISPLAY_NAME": null,
          "CONTEXT_ENTRYTYPE_DISPLAY_NAME": null
        },
        "DIRECT_VALID_FROM": "/Date(1515715200000)/",
        "DIRECT_VALID_TO": "/Date(1549065599000)/",
        "ASSIGNED_AT": null,
        "ASSIGNED_BY": null,
        "STATUS": 0,
        "REFERENCED_ENTRY": {
          "__deferred": {

```

```

        "uri": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=62,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')/REFERENCED_ENTRY"
    },
    "CONTEXT_ENTRY": {
        "__deferred": {
            "uri": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=62,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')/CONTEXT_ENTRY"
        }
    },
    "HIERARCHY": {
        "__deferred": {
            "uri": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=62,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')/HIERARCHY"
        }
    },
    "ASSIGNMENT_HISTORY": {
        "__deferred": {
            "uri": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=62,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')/ASSIGNMENT_HISTORY"
        }
    }
},
{
    "__metadata": {
        "id": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=64,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')",
        "uri": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=64,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')",
        "type": "IdmRestModel.ER_MX_ASSIGNMENT"
    },
    "TASK_GUID": "a56db5fe-da5a-4109-8518-054b7e0fa454",
    "LINK_ID": 64,
    "REFERENCED_ID": 13,
    "REFERENCED_MSKEYVALUE": "MX_PRIV:APPROVALS:READONLY",
    "REFERENCED_DISPLAY_NAME": "MX_PRIV:APPROVALS:READONLY",
    "REFERENCED_DESCRIPTION": null,
    "REFERENCED_ENTRYTYPE": "MX_PRIVILEGE",
    "VALID_FROM": "/Date(1515715200000)/",
    "VALID_TO": "/Date(1549065599000)/",
    "REASON": null,
    "STATE": 1,
    "STATE_TEXT": "OK",
    "DIRECTLY_ASSIGNED": false,
    "INHERITED": true,
    "OPERATION": "0",
    "CONTEXT": {
        "__metadata": {
            "type": "IdmRestModel.CONTEXT"
        },
        "CONTEXT_ID": null,
        "CONTEXT_DISPLAY_NAME": null,
        "CONTEXT_ENTRYTYPE_DISPLAY_NAME": null
    },
    "DIRECT_VALID_FROM": null,
    "DIRECT_VALID_TO": null,
    "ASSIGNED_AT": null,
    "ASSIGNED_BY": null,
    "STATUS": 0,
    "REFERENCED_ENTRY": {
        "__deferred": {

```

```

        "uri": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=64,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')/REFERENCED_ENTRY"
    },
    "CONTEXT_ENTRY": {
        "__deferred": {
            "uri": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=64,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')/CONTEXT_ENTRY"
        }
    },
    "HIERARCHY": {
        "__deferred": {
            "uri": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=64,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')/HIERARCHY"
        }
    },
    "ASSIGNMENT_HISTORY": {
        "__deferred": {
            "uri": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=64,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')/ASSIGNMENT_HISTORY"
        }
    }
},
{
    "__metadata": {
        "id": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=63,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')",
        "uri": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=63,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')",
        "type": "IdmRestModel.ER_MX_ASSIGNMENT"
    },
    "TASK_GUID": "a56db5fe-da5a-4109-8518-054b7e0fa454",
    "LINK_ID": 63,
    "REFERENCED_ID": 9,
    "REFERENCED_MSKEYVALUE": "MX_PRIV:CONFIG_R",
    "REFERENCED_DISPLAY_NAME": "MX_PRIV:CONFIG_R",
    "REFERENCED_DESCRIPTION": null,
    "REFERENCED_ENTRYTYPE": "MX_PRIVILEGE",
    "VALID_FROM": "/Date(1515715200000)/",
    "VALID_TO": "/Date(1549065599000)/",
    "REASON": null,
    "STATE": 1,
    "STATE_TEXT": "OK",
    "DIRECTLY_ASSIGNED": false,
    "INHERITED": true,
    "OPERATION": "0",
    "CONTEXT": {
        "__metadata": {
            "type": "IdmRestModel.CONTEXT"
        },
        "CONTEXT_ID": null,
        "CONTEXT_DISPLAY_NAME": null,
        "CONTEXT_ENTRYTYPE_DISPLAY_NAME": null
    },
    "DIRECT_VALID_FROM": null,
    "DIRECT_VALID_TO": null,
    "ASSIGNED_AT": null,
    "ASSIGNED_BY": null,
    "STATUS": 0,
    "REFERENCED_ENTRY": {
        "__deferred": {

```

```

        "uri": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=63,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')/REFERENCED_ENTRY"
    },
    "CONTEXT_ENTRY": {
        "__deferred": {
            "uri": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=63,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')/CONTEXT_ENTRY"
        }
    },
    "HIERARCHY": {
        "__deferred": {
            "uri": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=63,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')/HIERARCHY"
        }
    },
    "ASSIGNMENT_HISTORY": {
        "__deferred": {
            "uri": "https://./
idmrestapi/v2/service/ER_MX_ASSIGNMENT(LINK_ID=63,TASK_GUID=guid'a56db5fe-
da5a-4109-8518-054b7e0fa454')/ASSIGNMENT_HISTORY"
        }
    }
}
],
}
}

```

Related Information

<http://www.odata.org/developers/protocols/json-format#RepresentingTheRawValueOfAProperty> ➔
<http://www.odata.org/media/16352/%5Bms-odata%5D.pdf> ➔

8.12.8 Add Attribute Values to an Entry

Adds a single value or reference for a single attribute of an entry, specified by the entry's ID, the GUID of the task containing the attribute, and the attribute name.

The specified attribute must not be read-only and has to be exposed by the task, passed via the TaskGUID. The task must be a UI task and the currently logged-in user must be authorized to access this task.

ⓘ Note

You can only add values to multi-value attributes or entry reference attributes.

Single Value Entry Reference Attributes

Entry reference attributes not marked as multi-value attributes are represented the same way as multi-value entry reference attributes. However, if the attribute already has one entry reference, adding another entry reference fails.

Adding Multiple Values or References

To add multiple values to a multi-value attribute or multiple references to an entry reference attribute, specify the items in an array list.

Adding Binary Data

Media link attributes are filled in the same way as single value attributes. Insert the binary file content into the body of the POST request. Set the content type of the request accordingly. For example, if you post the content of a GIF image, you have to set the content type of the request to `image.gif`. To specify a file name, add a header with key `SLug` and the file name as its value to the request. If you do not pass the file name with the `SLug` header, a file name is computed based on the content type. For example, the content type `image/gif` results in a computed file name `image/gif`.

Note

Do not use an image upload based on HTML form data submit (`<form ... enctype="multipart/form-data"><input ... type="file">` ...). If you do, it is impossible to pass the mandatory XSRF-token.

By default, the size of the uploaded file is limited to 2097152 bytes (2 MB). The upload of larger files is blocked and results in `HTTPS response 403`. To allow the upload of larger files, adjust the value of the `v2.MaximumStorableFileSize` application property as necessary.

URI

```
(* [POST] *)
Multi value attribute:
-> Attribute is a collection of entries of type "MultiValueString",
"MultiValueDouble", or "MultiValueDateTime"
- Add one or more values:
URI      = './idmrestapi/v2/service/
ET_'EntryType'(ID='Mskey',TASK_GUID='TaskGUID')/MV_'AttributeName';
```

```
Entry reference attribute:
-> Attribute is a collection of entries of type "EntryReference"
- Add one or more references:
```

```
URI      = './idmrestapi/v2/service/  
ET_'EntryType'(ID='Mskey',TASK_GUID='TaskGUID')/ER_'AttributeName';
```

```
EntryType      = Edm.String;  
TaskGUID       = Edm.Guid;  
Mskey          = Edm.Int32;  
AttributeName  =      Edm.String;
```

```
Binary attribute:  
URI      = './idmrestapi/v2/service/  
ET_'EntryType'(ID='Mskey',TASK_GUID='TaskGUID')/BI_'AttributeName';  
URI      = './idmrestapi/v2/service/  
BI_'AttributeName'(ID='Mskey',TASK_GUID='TaskGUID')
```

Request Parameters

No request parameters. But a document defining the new attribute value or entry reference has to be passed.

Request Example 1

```
POST/idmrestapi/v2/service/  
ET_MX_PERSON(ID=7894,TASK_GUID=guid'D092EA23-9E5D-4FC7-9467-1E1D0BBE3629')/  
MV_MX_PHONE_ADDITIONAL HTTPS/1.1
```

```
[{  
    "VALUE":"+4989123456"  
}, {  
    "VALUE":"+4977664433"  
}]
```

Request Example 2

```
POST/idmrestapi/v2/service/  
ET_MX_PERSON(ID=7894,TASK_GUID=guid'D092EA23-9E5D-4FC7-9467-1E1D0BBE3629')/  
ER_MXREF_MX_ROLE HTTPS/1.1
```

```
{  
    "REFERENCED_ID":40814,  
    "CONTEXT":{  
        "ID":32145  
    },  
    "REASON":"User is Project Lead",  
    "VALID_FROM":"\\/Date(1358467200000)\\/",  
    "VALID_TO":"\\/Date(1369871999000)\\/"  
}
```

Response Types

- Content-Type: application/json | application/atom+xml

HTTPS Response Codes

- 201 - Created
The creation of the attribute value or reference was successful. The created entity is returned in the response.
- 401 - Unauthorized
If the user is not authenticated.
- 403 - Forbidden
The creation of the value or reference is not allowed for various reasons. Possible reasons are:
 - Missing mandatory parameter.
 - Conflicting unique attribute, that is, the value or reference already exists.
 - Conflicting assignment range.
 - Attribute is a single value entry reference and it already has one reference.The error code (if available) and the error message from SAP Identity Management and/or the database are returned in the response body.
- 404 - Not found
The specified task or entry was not found.

Response Example 1

HTTPS/1.1 201 Created

```
{ "d": {  
  "results": {  
    "__metadata": {  
      "uri": "/idmrestapi/v2/service/  
MV_MX_PHONE_ADDITIONAL(ID=7894, TASK_GUID=guid'D092EA23-9E5D-4FC7-9467-1E1D0BBE362  
9', VALUE='+4989123456')",  
      "type": "MultiValueString",  
    },  
    "VALUE": "+4989123456"  
  },  
}  
}}
```

Response Example 2

HTTPS/1.1 201 Created

```
{
  "d": {
    "results": {
      "__metadata": {
        "uri": "/idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=4711,TASK_GUID=guid'D092EA23-9E5D-4FC7-9467-1E1D0BBE3629')",
        "type": "EntryReference",
      },
      "LINK_ID": 4711,
      "REFERENCED_ID": 40814,
      "REFERENCED_DISPLAY_NAME": "Project Role",
      "VALID_FROM": "\\Date(1358467200000)\\",
      "VALID_TO": "\\Date(1369871999000)\\",
      "DIRECTLY_ASSIGNED": true,
      "INHERITED": false,
      "CONTEXT": {
        "ID": 32145,
        "DISPLAY_NAME": "ProjectX",
        "ENTRYTYPE_DISPLAY_NAME": "PROJECT"
      }
    }
  }
}
```

8.12.9 Update Attribute Values of an Entry

Updates the value for a single attribute of an entry, specified by the entry's ID, the GUID of the task containing the attribute, and the attribute name.

The specified attribute must not be read-only and has to be exposed by the task, passed via the TaskGUID. The task must be a UI task, and the currently logged-in user must be authorized to access this task.

Note

Updating single attribute values (except media data) is not supported in the initial version of the REST interface. This is only possible by updating the entries attributes, see *Update an Entry*.

Update Attributes with Value Help

For attributes having a value help, you must specify the value key as the value. If you specify the value text, an error occurs.

URI

```
(* [POST with MERGE method tunneling] *)
```

```
Single value attribute:
URI = './idmrestapi/v2/service/SV_EntryType'(ID='Mskey',TASK_GUID='TaskGUID')/'
```



```
AttributeName [ '/'$value' ];
```

Note

When using the REST API v 2 to create or update an entry with attribute of *Data Type Numeric* and *Presentation SingleLine*, note that the attribute value is saved in the Identity Management database in double format. For example: 1.0

```
Multi value attribute:  
Attribute is a collection of entries of type "MultiValueString",  
"MultiVaueDouble", "MultiValueDate"  
- Update one value:  
URI = './idmrestapi/v2/service/  
MV_'AttributeName'(ID='Mskey',TASK_GUID='TaskGUID', VALUE='Value')';
```

```
Entry reference attribute:  
Attribute is a collection of entries of type "EntryReference"  
- Update one reference:  
URI = './idmrestapi/v2/service/  
ER_'AttributeName'(ID='Mskey',TASK_GUID='TaskGUID', LINK_ID='LinkId');
```

```
EntryType      = Edm.String;  
TaskGUID       = Edm.Guid;  
Mskey          = Edm.Int32;  
AttributeName  = Edm.String;  
Value          = Edm.String | Edm.Double | Edm.DateTime;  
LinkId         = Edm.Int32;
```

```
Media attribute  
URI = './idmrestapi/v2/service/  
ET_'EntryType'(ID='Mskey',TASK_GUID='TaskGUID')/BI_' AttributeName;  
URI = './idmrestapi/v2/service/  
BI_'AttributeName'(ID='Mskey',TASK_GUID='TaskGUID')
```

Request Parameters

No request parameters. But a document defining the specified entry attribute has to be passed.

Response Types

None.

HTTPS Response Codes

- 204 - No Content
The update of the entry attribute was successful. No response body is returned.

Response Example

```
HTTPS/1.1 204 No Content
```

Related Information

[Add Attribute Values to an Entry \[page 878\]](#)

[Update an Entry \[page 863\]](#)

8.12.10 Search Entry References

There are several ways to search for references of an Identity Management attribute of the `Entry` reference type.

To search for entry references available at a certain entry, for instance, role assignments for a certain user, you can use custom query options. Using these query options, you can perform the following searches:

- **Status Search**
You can search by the status of the entry reference. This search type can be combined with either basic search or advanced search.
- **Basic Search**
You can perform a basic search on the specific attributes (`MSKEYVALUE` and `DISPLAYNAME`) of the referenced entry, for instance, the `DISPLAYNAME` of the referenced role.
- **Advanced Search**
You can perform an advanced search on the `DISPLAYNAME` attribute of a referenced entry.

ⓘ Note

If access limitations are specified on an entry type (for example: `MX_ROLE` or `MX_PRIVILEGE`), only the values of entry reference attributes (for example: `MXREF_MX_ROLE` or `MXREF_MX_PRIVILEGE`) that comply with these access limitations will be returned for the given user.

However, if you want to retrieve direct and indirect assignments of the user (using the `ER_MX_ASSIGNMENT` attribute), although access limitations are specified on an entry type, all assignments for the given user will be returned.

URIs

```
(* [GET] *)
- Direct entry reference search:
URI = './idmrestapi/v2/service/ER_' AttributeName '?ID=' Mskey '&TASK_GUID='
TaskGUID
- Search entry references for an entry attribute:
```

```

URI      = './idmrestapi/v2/service/ET_' EntryTypeName '(ID=' Mskey ',TASK_GUID='
TaskGUID ')/ER_'
      AttributeName
- Get an entry and search for an expanded entry reference attribute:
URI      = './idmrestapi/v2/service/ET_' EntryTypeName '(ID=' Mskey ',TASK_GUID='
TaskGUID
      ')?$expand=ER_' AttributeName
AttributeName = Name of an attribute that has the data type "Entry reference" in
the Identity Store schema;
EntryTypeName = Name of an entry type (MX_ENTRYTYPE) that exists in the Identity
Store schema;
Mskey        = Edm.Int32; (* ID of an entry *)
TaskGUID     = Edm.Guid;

```

Request Parameters

Status Search

```

Parameter      = [ StatusFilter ]
StatusFilter    = 'status=' StatusValue;
StatusValue     = 'ASSIGNED' | 'NOT_ASSIGNED' | 'FUTURE' | 'ASSIGNED_AND_PENDING'

```

- If no status filter and basic filter is specified, all assignments are returned.
- Status "ASSIGNED": All currently active assignments.
- Status "NOT_ASSIGNED":
 - Pending: Assignments are currently (de)provisioned or not yet approved. This explicitly does not include future assignments.
 - The assignment failed.
 - The assignment was rejected.
- Status "FUTURE": Pending assignments, because the valid-from date is still in the future.
- Status "ASSIGNED AND PENDING":
 - Assigned OK
 - Pending: Waiting for approval
 - Pending: Assignments are currently provisioned.
 - Future
 - Fails: waiting for another provisioning attempt
 - Rejected is excluded
- Status "HISTORY": All historical assignments.
 - All deleted assignments
 - All expired assignments

Basic Search

```

Parameter      = [ BasicFilter ] [ StatusFilter ]
BasicFilter     = 'filterBasic=' SearchValue;

```

```
SearchValue      = Complete or part (with wildcard character star [*]) of the
DISPLAYNAME or MSKEYVALUE
StatusFilter     = 'status=' StatusValue;
StatusValue      = 'ASSIGNED' | 'NOT_ASSIGNED' | 'FUTURE' | 'ASSIGNED_AND_PENDING'
```

Note

The wildcard character star [*] is allowed in the value of "filterBasic".

Advanced Search

```
Parameter        = [ StatusFilter ] [ Filter ]
StatusFilter      = 'status=' StatusValue;
StatusValue       = 'ASSIGNED' | 'NOT_ASSIGNED' | 'FUTURE' |
'ASSIGNED_AND_PENDING'
Filter            = '$filter=' DisplayNameFilter;
DisplayNameFilter  = 'REFERENCED_DISPLAY_NAME'ComparatorSearchValue
Comparator        = 'eq';
SearchValue       = Complete or part (with wildcard character star
[*]) of the DISPLAYNAME in single quotes
```

Note

- Currently, only the search for the referenced display name is supported.
- The wildcard character star [*] is allowed in the search value of \$filter.

Request Example 1

```
GET /idmrestapi/v2/service/ER_MXREF_MX_ROLE?
ID=25&TASK_GUID=guid'860E8B87-59F6-4FA6-A23D-B444D05816E4'&status=FUTURE
```

Request Example 2

```
GET /idmrestapi/v2/service/ET_MX_PERSON(ID=25,TASK_GUID=guid'860E8B87-59F6-4FA6-
A23D-B444D05816E4')/ER_MXREF_MX_ROLE?status=FUTURE&filterBasic=Q*
```

Request Example 3

```
GET /idmrestapi/v2/service/ET_MX_PERSON(ID=25,TASK_GUID=guid'860E8B87-59F6-4FA6-
A23D-B444D05816E4')?$expand=ER_MXREF_MX_ROLE
```

Request Example 4

```
GET /idmrestapi/v2/service/ER_MXREF_MX_ROLE?
ID=25&TASK_GUID=guid'860E8B87-59F6-4FA6-A23D-
B444D05816E4'&$filter=REFERENCED_DISPLAY_NAME eq 'A*.'&status=ASSIGNED
```

Response Types

Content-Type: application/json | application/atom+xml

HTTPS Status Codes:

- **200 OK**
Response Body: Collection of entities with the type EntryReference.
- **400 Bad Request**
 - The "status" parameter does not have one of the predefined values.
 - The given EntryTypeName does not fit to the given MSKEY.
 - The "filter" parameter does not have one of the predefined values.
- **403 Forbidden**
Logged-in user is not allowed to search for entry references.
- **404 Not Found**
AttributeName, EntryTypeName, Mskey, or TaskGUID does not exist.

Response Example 1 and 2

```
{
  "d": {
    "results": [
      {
        "__metadata": {
          "id": "https://./idmrestapi/v2/
service/ER_MXREF_MX_ROLE(LINK_ID=51,TASK_GUID=guid'860e8b87-59f6-4fa6-a23d-
b444d05816e4')",
          "uri": "https://./idmrestapi/v2/
service/ER_MXREF_MX_ROLE(LINK_ID=51,TASK_GUID=guid'860e8b87-59f6-4fa6-a23d-
b444d05816e4')",
          "type": "IdmRestModel.ER_MXREF_MX_ROLEET_MX_ROLEEntryReference"
        },
        "TASK_GUID": "860e8b87-59f6-4fa6-a23d-b444d05816e4",
        "LINK_ID": 51,
        "REFERENCED_ID": 30,
        "REFERENCED_MSKEYVALUE": "QA Role",
        "REFERENCED_DISPLAY_NAME": "QA Role",
        "REFERENCED_DESCRIPTION": null,
        "REFERENCED_ENTRYTYPE": "MX_ROLE",
        "VALID_FROM": "/Date(1515974400000)/",
        "VALID_TO": "/Date(1549065599000)/",
        "REASON": "This user needs the QA Role.",
        "STATE": 513,
        "STATE_TEXT": "Pending",
        "DIRECTLY_ASSIGNED": true,

```

```

    "INHERITED": false,
    "OPERATION": "0",
    "CONTEXT": {
      "__metadata": {
        "type": "IdMRestModel.CONTEXT"
      },
      "CONTEXT_ID": null,
      "CONTEXT_DISPLAY_NAME": null,
      "CONTEXT_ENTRYTYPE_DISPLAY_NAME": null
    },
    "DIRECT_VALID_FROM": "/Date(1515974400000)/",
    "DIRECT_VALID_TO": "/Date(1549065599000)/",
    "ASSIGNED_AT": null,
    "ASSIGNED_BY": null,
    "STATUS": 4,
    "REFERENCED_ENTRY": {
      "__deferred": {
        "uri": "https://./idmrestapi/v2/
service/ER_MXREF_MX_ROLE(LINK_ID=51,TASK_GUID=guid'860e8b87-59f6-4fa6-a23d-
b444d05816e4')/REFERENCED_ENTRY"
      }
    },
    "CONTEXT_ENTRY": {
      "__deferred": {
        "uri": "https://./idmrestapi/v2/
service/ER_MXREF_MX_ROLE(LINK_ID=51,TASK_GUID=guid'860e8b87-59f6-4fa6-a23d-
b444d05816e4')/CONTEXT_ENTRY"
      }
    },
    "HIERARCHY": {
      "__deferred": {
        "uri": "https://./idmrestapi/v2/
service/ER_MXREF_MX_ROLE(LINK_ID=51,TASK_GUID=guid'860e8b87-59f6-4fa6-a23d-
b444d05816e4')/HIERARCHY"
      }
    },
    "ASSIGNMENT_HISTORY": {
      "__deferred": {
        "uri": "https://./idmrestapi/v2/
service/ER_MXREF_MX_ROLE(LINK_ID=51,TASK_GUID=guid'860e8b87-59f6-4fa6-a23d-
b444d05816e4')/ASSIGNMENT_HISTORY"
      }
    }
  }
},
]
}

```

Response Example 3

```

{
  "d": {
    "__metadata": {
      "id": "https://./idmrestapi/v2/service/
ET_MX_PERSON(ID=23,TASK_GUID=guid'860e8b87-59f6-4fa6-a23d-b444d05816e4')",
      "uri": "https://./idmrestapi/v2/service/
ET_MX_PERSON(ID=23,TASK_GUID=guid'860e8b87-59f6-4fa6-a23d-b444d05816e4')",
      "type": "IdMRestModel.ET_MX_PERSON"
    },
    "ID": 25,
    "TASK_GUID": "860e8b87-59f6-4fa6-a23d-b444d05816e4",
    "SV_MSKEYVALUE": null,
    "SV_DISPLAYNAME": null,
  }
}

```

```

"SV_MX_LASTNAME": null,
"SV_MX_FIRSTNAME": null,
"ER_MXREF_MX_ROLE": {
  "results": [
    {
      "__metadata": {
        "id": "https://.//idmrestapi/v2/
service/ER_MXREF_MX_ROLE(LINK_ID=43,TASK_GUID=guid'860e8b87-59f6-4fa6-a23d-
b444d05816e4')",
        "uri": "https://.//idmrestapi/v2/
service/ER_MXREF_MX_ROLE(LINK_ID=43,TASK_GUID=guid'860e8b87-59f6-4fa6-a23d-
b444d05816e4')",
        "type":
"IdMRestModel.ER_MXREF_MX_ROLE.ENTRY_REFERENCE"
      },
      "TASK_GUID": "860e8b87-59f6-4fa6-a23d-b444d05816e4",
      "LINK_ID": 43,
      "REFERENCED_ID": 26,
      "REFERENCED_MSKEYVALUE": "AdminRole",
      "REFERENCED_DISPLAY_NAME": "Admin Role",
      "REFERENCED_DESCRIPTION": null,
      "REFERENCED_ENTRYTYPE": "MX_ROLE",
      "VALID_FROM": "/Date(1515542400000)/",
      "VALID_TO": "/Date(1548979199000)/",
      "REASON": "Administrator in Project X",
      "STATE": 1,
      "STATE_TEXT": "OK",
      "DIRECTLY_ASSIGNED": true,
      "INHERITED": false,
      "OPERATION": "0",
      "CONTEXT": {
        "__metadata": {
          "type": "IdMRestModel.CONTEXT"
        },
        "CONTEXT_ID": 25,
        "CONTEXT_DISPLAY_NAME": "Project X",
        "CONTEXT_ENTRYTYPE_DISPLAY_NAME": "Project"
      },
      "DIRECT_VALID_FROM": "/Date(1515542400000)/",
      "DIRECT_VALID_TO": "/Date(1548979199000)/",
      "ASSIGNED_AT": null,
      "ASSIGNED_BY": null,
      "STATUS": 0,
      "REFERENCED_ENTRY": {
        "__deferred": {
          "uri": "https://.//idmrestapi/v2/
service/ER_MXREF_MX_ROLE(LINK_ID=43,TASK_GUID=guid'860e8b87-59f6-4fa6-a23d-
b444d05816e4')/REFERENCED_ENTRY"
        }
      },
      "CONTEXT_ENTRY": {
        "__deferred": {
          "uri": "https://.//idmrestapi/v2/
service/ER_MXREF_MX_ROLE(LINK_ID=43,TASK_GUID=guid'860e8b87-59f6-4fa6-a23d-
b444d05816e4')/CONTEXT_ENTRY"
        }
      },
      "HIERARCHY": {
        "__deferred": {
          "uri": "https://.//idmrestapi/v2/
service/ER_MXREF_MX_ROLE(LINK_ID=43,TASK_GUID=guid'860e8b87-59f6-4fa6-a23d-
b444d05816e4')/HIERARCHY"
        }
      },
      "ASSIGNMENT_HISTORY": {
        "__deferred": {

```

Response Example 4

890 PUBLIC


```

        "uri": "https://./idmrestapi/v2/
service/ER_MXREF_MX_ROLE(LINK_ID=43,TASK_GUID=guid'860e8b87-59f6-4fa6-a23d-
b444d05816e4')/CONTEXT_ENTRY"
    },
    "HIERARCHY": {
        "__deferred": {
            "uri": "https://./idmrestapi/v2/
service/ER_MXREF_MX_ROLE(LINK_ID=43,TASK_GUID=guid'860e8b87-59f6-4fa6-a23d-
b444d05816e4')/HIERARCHY"
        }
    },
    "ASSIGNMENT_HISTORY": {
        "__deferred": {
            "uri": "https://./idmrestapi/v2/
service/ER_MXREF_MX_ROLE(LINK_ID=43,TASK_GUID=guid'860e8b87-59f6-4fa6-a23d-
b444d05816e4')/ASSIGNMENT_HISTORY"
        }
    }
}
]
}
}

```

8.12.11 Get Entry Reference History

To get an entry reference history (that is, an assignment history), you send an HTTPS GET request to the endpoint of the entry reference attribute. For example, if you want to get the history of a role assignment, you send an HTTPS GET request to the /ER_MXREF_MX_ROLE endpoint and append the /ASSIGNMENT_HISTORY navigation.

Request

URI for getting the history of a role assignment: ' ./idmrestapi/v2/service/ER_' AttributeName '(LINK_ID=' Link ID ',TASK_GUID=' TaskGUID)/ASSIGNMENT_HISTORY

AttributeName = Name of an attribute with data type "Entry reference" in the Identity Store schema
Link ID = Link ID of the assignment
TaskGUID = TaskGUID of a task that is used for modifying an entry and that contains an entry reference attribute.
For example: Assign a role to a user

HTTPS Method: *GET*

Request Example

```

/idmrestapi/v2/service/
ER_MXREF_MX_ROLE(LINK_ID=62,TASK_GUID=guid'102E1D96-1813-4A58-9880-9572056F0FD1')
/ASSIGNMENT_HISTORY

```

Response

Response Status and Error Codes

Code	Reason	Description
200	OK	The response body contains the history of the assignment specified by the given LINK_ID.
400	Bad Request	Task is a create task.
403	Forbidden	Logged-in user is not allowed to access the task. Task is a self-service task and the MSKEY is not from the logged-in user.
404	Not Found	The specified LINK_ID does not exist. The specified TASK_GUID does not exist.

Response Example

```
{
  "d": {
    "results": [
      {
        "__metadata": {
          "id": "https://./idmrestapi/v2/service/
ASSIGNMENT_HISTORY(LINK_ID=62,TASK_GUID=guid'102e1d96-1813-4a58-9880-9572056f0fd1
')",
          "uri": "https://./idmrestapi/v2/service/
ASSIGNMENT_HISTORY(LINK_ID=62,TASK_GUID=guid'102e1d96-1813-4a58-9880-9572056f0fd1
')",
          "type": "IdMRestModel.AssignmentHistory"
        },
        "LINK_ID": 62,
        "TASK_GUID": "102e1d96-1813-4a58-9880-9572056f0fd1",
        "OPERATION": "Assignment requested",
        "ACTION_BY": "administrator",
        "ACTION_ON": "",
        "ACTION_AT": "/Date(1515772938587)/",
        "REASON": null
      },
      {
        "__metadata": {
          "id": "https://./idmrestapi/v2/service/
ASSIGNMENT_HISTORY(LINK_ID=62,TASK_GUID=guid'102e1d96-1813-4a58-9880-9572056f0fd1
')",
          "uri": "https://./idmrestapi/v2/service/
ASSIGNMENT_HISTORY(LINK_ID=62,TASK_GUID=guid'102e1d96-1813-4a58-9880-9572056f0fd1
')",
          "type": "IdMRestModel.AssignmentHistory"
        },
        "LINK_ID": 62,
        "TASK_GUID": "102e1d96-1813-4a58-9880-9572056f0fd1",
        "OPERATION": "Assignment OK",
        "ACTION_BY": "<System>",
        "ACTION_ON": "",
        "ACTION_AT": "/Date(1515772944133)/",
        "REASON": null
      }
    ]
  }
}
```

```
}  
}
```

8.12.12 Get Entry Reference Hierarchy

To get an entry reference hierarchy (that is, an assignment hierarchy), you send an HTTPS GET request to the endpoint of the entry reference attribute. For example, if you want to get the hierarchy of a role assignment, you send an HTTPS GET request to the `/ER_MXREF_MX_ROLE` endpoint and append the `/HIERARCHY` navigation.

Only the first-level children of the assignment are returned in the response body. For example, if a person has a role assignment that contains privilege 1 and privilege 2, the response body will return the assignments of privilege 1 and privilege 2 to that person.

Only the values of the following attributes are returned in the response body: `REFERENCED_ID`, `REFERENCED_MSKEYVALUE`, `REFERENCED_DISPLAY_NAME`, `REFERENCED_ENTRYTYPE`, `VALID_FROM` and `VALID_TO`.

Request

URI for getting the hierarchy of a role assignment: `'./idmrestapi/v2/service/ER_'` `AttributeName` `'(LINK_ID='` `Link ID` `',TASK_GUID='` `TaskGUID``)/HIERARCHY`

```
AttributeName = Name of an attribute with data type "Entry reference" in the  
Identity Store schema  
Link ID       = Link ID of the assignment  
TaskGUID      = TaskGUID of a task that is used for modifying an entry and that  
contains an entry reference attribute.  
                For example: Assign a role to a user
```

HTTPS Method: [GET](#)

Request Example

```
/idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=26851,TASK_GUID=guid'E5757912-  
A669-4F3B-83CF-75381194F0BC')/HIERARCHY
```

Response

Response Status and Error Codes

Code	Reason	Description
200	OK	The response body contains the hierarchy of the assignment specified by the given <code>LINK_ID</code> .

Code	Reason	Description
400	Bad Request	Task is a create task.
403	Forbidden	Logged-in user is not allowed to access the task. Task is a self-service task and the MSKEY is not from the logged-in user.
404	Not Found	The specified LINK_ID does not exist. The specified TASK_GUID does not exist.

Response Example

```
{
  "d":{
    "results":[
      {
        "__metadata":{
          "id":"https://./
idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=26857,TASK_GUID=guid'e5757912-
a669-4f3b-83cf-75381194f0bc')",
          "uri":"https://./
idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=26857,TASK_GUID=guid'e5757912-
a669-4f3b-83cf-75381194f0bc')",
          "type":"IdmRestModel.ER_MXREF_MX_ROLE.ER_MX_ROLEEntryReference"
        },
        "TASK_GUID":"e5757912-a669-4f3b-83cf-75381194f0bc",
        "LINK_ID":26857,
        "REFERENCED_ID":7062,
        "REFERENCED_MSKEYVALUE":"ChildRole2",
        "REFERENCED_DISPLAY_NAME":"ChildRole2",
        "REFERENCED_DESCRIPTION":null,
        "REFERENCED_ENTRYTYPE":"MX_ROLE",
        "VALID_FROM":"/Date(1550016000000)/",
        "VALID_TO":"/Date(1550361599000)/",
        "REASON":null,
        "STATE":1,
        "STATE_TEXT":"OK",
        "DIRECTLY_ASSIGNED":false,
        "INHERITED":true,
        "OPERATION":null,
        "CONTEXT":{
          "__metadata":{
            "type":"IdmRestModel.CONTEXT"
          },
          "CONTEXT_ID":null,
          "CONTEXT_DISPLAY_NAME":null,
          "CONTEXT_ENTRYTYPE_DISPLAY_NAME":null
        },
        "DIRECT_VALID_FROM":null,
        "DIRECT_VALID_TO":null,
        "CURRENT_VALID_FROM":null,
        "CURRENT_VALID_TO":null,
        "ASSIGNED_AT":null,
        "ASSIGNED_BY":null,
        "STATUS":null,
        "CONTEXT_MSKEY":null,
        "CONTEXT_MSKEYVALUE":null,
        "REFERENCED_ENTRY":{
          "__deferred":{
            "uri":"https://./
idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=26857,TASK_GUID=guid'e5757912-
a669-4f3b-83cf-75381194f0bc')/REFERENCED_ENTRY"
```

```

    }
  },
  "CONTEXT_ENTRY": {
    "__deferred": {
      "uri": "https://./
idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=26857,TASK_GUID=guid'e5757912-
a669-4f3b-83cf-75381194f0bc')/CONTEXT_ENTRY"
    }
  },
  "HIERARCHY": {
    "__deferred": {
      "uri": "https://./
idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=26857,TASK_GUID=guid'e5757912-
a669-4f3b-83cf-75381194f0bc')/HIERARCHY"
    }
  },
  "ASSIGNMENT_HISTORY": {
    "__deferred": {
      "uri": "https://./
idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=26857,TASK_GUID=guid'e5757912-
a669-4f3b-83cf-75381194f0bc')/ASSIGNMENT_HISTORY"
    }
  }
},
{
  "__metadata": {
    "id": "https://./
idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=26850,TASK_GUID=guid'e5757912-
a669-4f3b-83cf-75381194f0bc')",
    "uri": "https://./
idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=26850,TASK_GUID=guid'e5757912-
a669-4f3b-83cf-75381194f0bc')",
    "type": "IdMRestModel.ER_MXREF_MX_ROLE.ET_MX_ROLEEntryReference"
  },
  "TASK_GUID": "e5757912-a669-4f3b-83cf-75381194f0bc",
  "LINK_ID": 26850,
  "REFERENCED_ID": 7061,
  "REFERENCED_MSKEYVALUE": "ChildRole",
  "REFERENCED_DISPLAY_NAME": "ChildRole",
  "REFERENCED_DESCRIPTION": null,
  "REFERENCED_ENTRYTYPE": "MX_ROLE",
  "VALID_FROM": "/Date(1550016000000)/",
  "VALID_TO": "/Date(1550361599000)/",
  "REASON": null,
  "STATE": 1,
  "STATE_TEXT": "OK",
  "DIRECTLY_ASSIGNED": false,
  "INHERITED": true,
  "OPERATION": null,
  "CONTEXT": {
    "__metadata": {
      "type": "IdMRestModel.CONTEXT"
    },
    "CONTEXT_ID": null,
    "CONTEXT_DISPLAY_NAME": null,
    "CONTEXT_ENTRYTYPE_DISPLAY_NAME": null
  },
  "DIRECT_VALID_FROM": null,
  "DIRECT_VALID_TO": null,
  "CURRENT_VALID_FROM": null,
  "CURRENT_VALID_TO": null,
  "ASSIGNED_AT": null,
  "ASSIGNED_BY": null,
  "STATUS": null,
  "CONTEXT_MSKEY": null,
  "CONTEXT_MSKEYVALUE": null,
  "REFERENCED_ENTRY": {
    "__deferred": {

```

```

        "uri": "https://./
idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=26850, TASK_GUID=guid'e5757912-
a669-4f3b-83cf-75381194f0bc')/REFERENCED_ENTRY"
    },
    "CONTEXT_ENTRY": {
        "__deferred": {
            "uri": "https://./
idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=26850, TASK_GUID=guid'e5757912-
a669-4f3b-83cf-75381194f0bc')/CONTEXT_ENTRY"
        }
    },
    "HIERARCHY": {
        "__deferred": {
            "uri": "https://./
idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=26850, TASK_GUID=guid'e5757912-
a669-4f3b-83cf-75381194f0bc')/HIERARCHY"
        }
    },
    "ASSIGNMENT_HISTORY": {
        "__deferred": {
            "uri": "https://./
idmrestapi/v2/service/ER_MXREF_MX_ROLE(LINK_ID=26850, TASK_GUID=guid'e5757912-
a669-4f3b-83cf-75381194f0bc')/ASSIGNMENT_HISTORY"
        }
    }
}
    ]
}
}

```

8.12.13 Context Usage

The assignment of privileges and roles supports the specification of a context. The same role or privilege can be assigned with different contexts. Additionally, it can also be assigned without a context. If an assignment is done without a context, the value for the context must be `null`.

8.12.14 Virus Scanning

The Virus Scan Interface of the AS Java can scan binary data for viruses during the upload.

The following prerequisites must be met:

- A third-party virus scanner is installed. As it is not part of the SAP NetWeaver AS for Java it has to be licensed separately.
- A default profile must be configured in the Virus Scan Interface (VSI) infrastructure.

During the deployment of the REST application, two VSI profiles are created, unless these profiles have already been created during a former deployment:

- The `idmRest_ImageUpload` profile used for binary attributes with the `Picture` presentation type. The profile contains a default configuration for virus scanning, MIME type matching (tests if the file content matches the file name extension) and content type filtering based on a MIME type allowlist.

- The `idmRest_FileUpload` used for any other binary attributes.
The profile is a default configuration for virus scanning, MIME type matching and active content blocking.
This prevents, for example, uploading XMLs that contain Javascript.

The profiles must be activated manually in the SAP NetWeaver Administrator, otherwise no checks are performed. If the VSI scan detects any security issue in the uploaded data, the request results in a **Forbidden** response with a short description of the reason.

For more information, see the Virus Scan Interface documentation in the SAP Library.

Related Information

[Java-specific Configuration of the Virus Scan Interface](#)

8.13 Entry Type Access URIs

You can use the URIs described in this section to access entry types that are defined in the Identity Store schema (for example, `MX_PERSON`, `MX_ROLE`) or custom entry types (for example, `PROJECT`). You can get the following information about the entry types:

- Searchability
This entry type is searchable from the Identity Management User Interface.
- Context entry type
This entry type is defined as context entry type.
- Display Form
This entry type contains a display form that is used to display details about entries of this type in the Identity Management User Interface
- Search Form
This entry type contains a search form that is used to search for entries of this type in the Identity Management User Interface
- List Attributes
This entry type contains list attributes, that is, attributes that are included in the search result in the Identity Management User Interface.

Related Information

[Get All Entry Types \[page 898\]](#)

[Get a Specific Entry Type \[page 903\]](#)

8.13.1 Get All Entry Types

To get all entry types, you send an HTTPS GET request to the /ENTRY_TYPES endpoint.

Request

URI: `./ idmrestapi/v2/service/ENTRY_TYPES`

HTTPS Method: *GET*

Request Parameters

Parameter	Description
\$expand	To inline deferred display task, search task and list attributes.
\$filter	To perform a search based on a property. The supported filter properties for entry types are the following: IS_SEARCHABLE and IS_CONTEXT_TYPE.

Request Example 1

```
./ idmrestapi/v2/service/ENTRY_TYPES
```

Request Example 2

```
./idmrestapi/v2/service/ENTRY_TYPES?$filter=IS_SEARCHABLE eq false
```

Request Example 3

```
./idmrestapi/v2/service/ENTRY_TYPES?$filter=IS_CONTEXT_TYPE eq true
```

Request Example 4

```
./idmrestapi/v2/service/ENTRY_TYPES?$filter=IS_CONTEXT_TYPE eq true and  
IS_SEARCHABLE eq true
```

Request Example 5

```
./idmrestapi/v2/service/ENTRY_TYPES?$filter=IS_CONTEXT_TYPE eq true and  
IS_SEARCHABLE eq true&$expand=LIST_ATTRIBUTES
```

Response Types

```
Content-Type: application/json | application/atom+xml
```


Response

Response Status and Error Codes

Code	Reason	Description
200	OK	Response Body: Feed containing entry types
400	Bad Request	The filter parameter does not have one of the predefined values.
403	Forbidden	Logged in user does not have access to all entry types.

Response Example 1

```
{
  "d": {
    "results": [
      {
        "__metadata": {
          "id": "https://./idmrestapi/v2/service/ENTRY_TYPES('MX_PERSON')",
          "uri": "https://./idmrestapi/v2/service/ENTRY_TYPES('MX_PERSON')",
          "type": "IdmRestModel.ENTRY_TYPES"
        },
        "NAME": "MX_PERSON",
        "DISPLAY_NAME": "Person",
        "IS_CONTEXT_TYPE": false,
        "IS_SEARCHABLE": true,
        "DISPLAY_FORM": {
          "__deferred": {
            "uri": "https://./idmrestapi/v2/service/ENTRY_TYPES('MX_PERSON')/DISPLAY_FORM"
          }
        },
        "SEARCH_FORM": {
          "__deferred": {
            "uri": "https://./idmrestapi/v2/service/ENTRY_TYPES('MX_PERSON')/SEARCH_FORM"
          }
        },
        "LIST_ATTRIBUTES": {
          "__deferred": {
            "uri": "https://./idmrestapi/v2/service/ENTRY_TYPES('MX_PERSON')/LIST_ATTRIBUTES"
          }
        }
      }
    ]
  },
  {
    "__metadata": {
      "id": "https://./idmrestapi/v2/service/ENTRY_TYPES('MX_PENDING_VALUE')",
      "uri": "https://./idmrestapi/v2/service/ENTRY_TYPES('MX_PENDING_VALUE')",
      "type": "IdmRestModel.ENTRY_TYPES"
    },
    "NAME": "MX_PENDING_VALUE",
    "DISPLAY_NAME": "Pending Value",
    "IS_CONTEXT_TYPE": false,
    "IS_SEARCHABLE": false,
    "DISPLAY_FORM": {
      "__deferred": {

```

```

        "uri": "https://./idmrestapi/v2/service/
ENTRY_TYPES('MX_PENDING_VALUE')/DISPLAY_FORM"
    },
    "SEARCH_FORM": {
        "__deferred": {
            "uri": "https://./idmrestapi/v2/service/
ENTRY_TYPES('MX_PENDING_VALUE')/SEARCH_FORM"
        }
    },
    "LIST_ATTRIBUTES": {
        "__deferred": {
            "uri": "https://./idmrestapi/v2/service/
ENTRY_TYPES('MX_PENDING_VALUE')/LIST_ATTRIBUTES"
        }
    }
},
{
    "__metadata": {
        "id": "https://./idmrestapi/v2/service/ENTRY_TYPES('PROJECT')",
        "uri": "https://./idmrestapi/v2/service/ENTRY_TYPES('PROJECT')",
        "type": "IdmRestModel.ENTRY_TYPES"
    },
    "NAME": "PROJECT",
    "DISPLAY_NAME": "Project",
    "IS_CONTEXT_TYPE": true,
    "IS_SEARCHABLE": true,
    "DISPLAY_FORM": {
        "__deferred": {
            "uri": "https://./idmrestapi/v2/service/
ENTRY_TYPES('PROJECT')/DISPLAY_FORM"
        }
    },
    "SEARCH_FORM": {
        "__deferred": {
            "uri": "https://./idmrestapi/v2/service/
ENTRY_TYPES('PROJECT')/SEARCH_FORM"
        }
    },
    "LIST_ATTRIBUTES": {
        "__deferred": {
            "uri": "https://./idmrestapi/v2/service/
ENTRY_TYPES('PROJECT')/LIST_ATTRIBUTES"
        }
    }
}
],
}
}

```

Response Example 2

```

{
    "d": {
        "results": [
            {
                "__metadata": {
                    "id": "https://./idmrestapi/v2/service/
ENTRY_TYPES('MX_PENDING_VALUE')",
                    "uri": "https://./idmrestapi/v2/service/
ENTRY_TYPES('MX_PENDING_VALUE')",
                    "type": "IdmRestModel.ENTRY_TYPES"
                },
                "NAME": "MX_PENDING_VALUE",
                "DISPLAY_NAME": "Pending Value",
                "IS_CONTEXT_TYPE": false,
                "IS_SEARCHABLE": false,
            }
        ]
    }
}

```

```

        "DISPLAY_FORM": {
            "__deferred": {
                "uri": "https://./idmrestapi/v2/service/
ENTRY_TYPES('MX_PENDING_VALUE')/DISPLAY_FORM"
            }
        },
        "SEARCH_FORM": {
            "__deferred": {
                "uri": "https://./idmrestapi/v2/service/
ENTRY_TYPES('MX_PENDING_VALUE')/SEARCH_FORM"
            }
        },
        "LIST_ATTRIBUTES": {
            "__deferred": {
                "uri": "https://./idmrestapi/v2/service/
ENTRY_TYPES('MX_PENDING_VALUE')/LIST_ATTRIBUTES"
            }
        }
    }
}

```

Response Example 3 and 4

```

{
    "d": {
        "results": [
            {
                "__metadata": {
                    "id": "https://./idmrestapi/v2/service/ENTRY_TYPES('PROJECT')",
                    "uri": "https://./idmrestapi/v2/service/ENTRY_TYPES('PROJECT')",
                    "type": "IdmRestModel.ENTRY_TYPES"
                },
                "NAME": "PROJECT",
                "DISPLAY_NAME": "Project",
                "IS_CONTEXT_TYPE": true,
                "IS_SEARCHABLE": true,
                "DISPLAY_FORM": {
                    "__deferred": {
                        "uri": "https://./idmrestapi/v2/service/
ENTRY_TYPES('PROJECT')/DISPLAY_FORM"
                    }
                },
                "SEARCH_FORM": {
                    "__deferred": {
                        "uri": "https://./idmrestapi/v2/service/
ENTRY_TYPES('PROJECT')/SEARCH_FORM"
                    }
                },
                "LIST_ATTRIBUTES": {
                    "__deferred": {
                        "uri": "https://./idmrestapi/v2/service/
ENTRY_TYPES('PROJECT')/LIST_ATTRIBUTES"
                    }
                }
            }
        ]
    }
}

```

Response Example 5

```

{
    "d": {
        "results": [

```

```

{
  "__metadata": {
    "id": "https://./idmrestapi/v2/service/ENTRY_TYPES('PROJECT')",
    "uri": "https://./idmrestapi/v2/service/ENTRY_TYPES('PROJECT')",
    "type": "IdMRestModel.ENTRY_TYPES"
  },
  "NAME": "PROJECT",
  "DISPLAY_NAME": "Project",
  "IS_CONTEXT_TYPE": true,
  "IS_SEARCHABLE": true,
  "DISPLAY_FORM": {
    "__deferred": {
      "uri": "https://./idmrestapi/v2/service/ENTRY_TYPES('PROJECT')/DISPLAY_FORM"
    }
  },
  "SEARCH_FORM": {
    "__deferred": {
      "uri": "https://./idmrestapi/v2/service/ENTRY_TYPES('PROJECT')/SEARCH_FORM"
    }
  },
  "LIST_ATTRIBUTES": {
    "results": [
      {
        "__metadata": {
          "id": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'ffffffff-ffff-ffff-ffff-fffffffffffffff',ATTRIBUTE_NAME='DISPLAYNAME')",
          "uri": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'ffffffff-ffff-ffff-ffff-fffffffffffffff',ATTRIBUTE_NAME='DISPLAYNAME')",
          "type": "IdMRestModel.TaskAttribute"
        },
        "TASK_GUID": "ffffffff-ffff-ffff-ffff-fffffffffffffff",
        "ATTRIBUTE_NAME": "DISPLAYNAME",
        "DISPLAY_NAME": "Display Name",
        "DEFAULT_VALUE": null,
        "HIDE_INPUT": false,
        "TOOLTIP": null,
        "VALUEHELP": false,
        "NUMBER_OF_LINES": 0,
        "READONLY": false,
        "MANDATORY": false,
        "IS_MIXED_CASE": false,
        "IS_MIXED_LETTERS_NUMBERS": false,
        "IS_UNIQUE": false,
        "MIN_LENGTH": 0,
        "MAX_LENGTH": 0,
        "VALIDATION_REGEX": "",
        "CONFIRM_INPUT": false,
        "LIST_ENTRIES_ON_LOAD": false,
        "SHOW_SEARCH_FIELD": false,
        "PRESENTATION_TYPE": 1,
        "STORAGE_TYPE": 1,
        "IS_MULTIVALUE": false,
        "REFERENCED_ENTRYTYPE": null,
        "REFERENCED_ENTRYTYPE_DISPLAY_NAME": null,
        "ALLOWED_VALUES": {
          "__deferred": {
            "uri": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'ffffffff-ffff-ffff-ffff-fffffffffffffff',ATTRIBUTE_NAME='DISPLAYNAME')/ALLOWED_VALUES"
          }
        }
      }
    ]
  },
  "__metadata": {

```

```
"id": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'ffffffff-ffff-ffff-ffff-ffffffffffffff',ATTRIBUTE_NAME='MSKEYVALUE')",
"uri": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'ffffffff-ffff-ffff-ffff-ffffffffffffff',ATTRIBUTE_NAME='MSKEYVALUE')",
"type": "IdMRestModel.TaskAttribute"
},
{
  "TASK_GUID": "ffffffff-ffff-ffff-ffff-ffffffffffffff",
  "ATTRIBUTE_NAME": "MSKEYVALUE",
  "DISPLAY_NAME": "Unique ID",
  "DEFAULT_VALUE": null,
  "HIDE_INPUT": false,
  "TOOLTIP": null,
  "VALUEHELP": false,
  "NUMBER_OF_LINES": 0,
  "READONLY": false,
  "MANDATORY": false,
  "IS_MIXED_CASE": false,
  "IS_MIXED_LETTERS_NUMBERS": false,
  "IS_UNIQUE": false,
  "MIN_LENGTH": 0,
  "MAX_LENGTH": 0,
  "VALIDATION_REGEX": "",
  "CONFIRM_INPUT": false,
  "LIST_ENTRIES_ON_LOAD": false,
  "SHOW_SEARCH_FIELD": false,
  "PRESENTATION_TYPE": 1,
  "STORAGE_TYPE": 1,
  "IS_MULTIVALUE": false,
  "REFERENCED_ENTRYTYPE": null,
  "REFERENCED_ENTRYTYPE_DISPLAY_NAME": null,
  "ALLOWED_VALUES": {
    "__deferred": {
      "uri": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'ffffffff-ffff-ffff-ffff-ffffffffffffff',ATTRIBUTE_NAME='MSKEYVALUE')/ALLOWED_VALUES"
    }
  }
}
]
}
```

8.13.2 Get a Specific Entry Type

To get a specific entry type, you send an HTTPS GET request to the `/ENTRY_TYPES` endpoint with the entry type's name as a key. For example, if you want to get the `MX_PERSON` entry type, you send an HTTPS GET request to `./idmrestapi/v2/service/ENTRY_TYPES('MX_PERSON')`

The response body for each entry type contains information about the searchability of the entry type, whether it is a context entry types or not, the search and display forms defined for this entry type, as well as its list attributes.

The `$expand` parameter is supported, in case you want to expand the search form, the display form or the list attributes.

Since expanding the display or search form only returns the corresponding forms definition, “nested” expand is supported to get those forms task attributes within one request (see Request Example 3).

Request

URI: `./ idmrestapi/v2/service/ENTRY_TYPES(' Entry_Type_Name ')`

Entry_Type_Name = The name of the entry type

HTTPS Method: [GET](#)

Request Parameters

Parameter	Description
\$expand	To inline deferred display form, search form and list attributes.

Request Example 1

```
./idmrestapi/v2/service/ENTRY_TYPES('MX_PERSON')
```

Request Example 2

```
./idmrestapi/v2/service/ENTRY_TYPES('MX_PERSON')?$expand=DISPLAY_FORM,SEARCH_FORM
```

Request Example 3

```
./idmrestapi/v2/service/ENTRY_TYPES('PROJECT')?$expand=DISPLAY_FORM/  
TASK_ATTRIBUTES
```

Response Types

Content-Type: application/json | application/atom+xml

Response

Response Status and Error Codes

Code	Reason	Description
200	OK	Response Body: Entry type entry
403	Forbidden	Logged in user does not have access the concrete entry type.

Code	Reason	Description
404	Not Found	- The entry type name does not exist. - The "expand" parameter does not have one of the predefined values.

Response Example 1

```
{
  "d": {
    "__metadata": {
      "id": "https://.//idmrestapi/v2/service/ENTRY_TYPES('MX_PERSON')",
      "uri": "https://.//idmrestapi/v2/service/ENTRY_TYPES('MX_PERSON')",
      "type": "IdMRestModel.ENTRY_TYPES"
    },
    "NAME": "MX_PERSON",
    "DISPLAY_NAME": "Person",
    "IS_CONTEXT_TYPE": false,
    "IS_SEARCHABLE": true,
    "DISPLAY_FORM": {
      "__deferred": {
        "uri": "https://.//idmrestapi/v2/service/ENTRY_TYPES('MX_PERSON')/"
      }
    },
    "SEARCH_FORM": {
      "__deferred": {
        "uri": "https://.//idmrestapi/v2/service/ENTRY_TYPES('MX_PERSON')/"
      }
    },
    "LIST_ATTRIBUTES": {
      "__deferred": {
        "uri": "https://.//idmrestapi/v2/service/ENTRY_TYPES('MX_PERSON')/"
      }
    }
  }
}
```

Response Example 2

```
{
  "d": {
    "__metadata": {
      "id": "https://.//idmrestapi/v2/service/ENTRY_TYPES('MX_PERSON')",
      "uri": "https://.//idmrestapi/v2/service/ENTRY_TYPES('MX_PERSON')",
      "type": "IdMRestModel.ENTRY_TYPES"
    },
    "NAME": "MX_PERSON",
    "DISPLAY_NAME": "Person",
    "IS_CONTEXT_TYPE": false,
    "IS_SEARCHABLE": true,
    "DISPLAY_FORM": {
      "__metadata": {
        "id": "https://.//idmrestapi/v2/service/TASK_CONFIGURATIONS(guid'f26bba84-b838-4a14-8598-bf327b026e0a')",
        "uri": "https://.//idmrestapi/v2/service/TASK_CONFIGURATIONS(guid'f26bba84-b838-4a14-8598-bf327b026e0a')",
        "type": "IdMRestModel.TASK_CONFIGURATIONS"
      },
      "TASK_GUID": "f26bba84-b838-4a14-8598-bf327b026e0a",
      "TASK_ID": 477,
    }
  }
}
```

```

        "TASK_GROUP": null,
        "TASK_DISPLAY_NAME": "Display Identity",
        "TASK_HEADER": null,
        "TASK_FOOTER": null,
        "TASK_HAS_SUBMIT_BUTTON": false,
        "TASK_SUBMIT_BUTTON_CAPTION": null,
        "TASK_HAS_MODIFY_BUTTON": false,
        "TASK_MODIFY_BUTTON_CAPTION": null,
        "TASK_CONFIRMATION_MESSAGE": null,
        "TASK_ATTRIBUTES": {
            "__deferred": {
                "uri": "https://./idmrestapi/v2/service/
TASK_CONFIGURATIONS(guid'f26bba84-b838-4a14-8598-bf327b026e0a')/TASK_ATTRIBUTES"
            }
        },
        "CURRENT_ENTRY": {
            "__deferred": {
                "uri": "https://./idmrestapi/v2/service/
TASK_CONFIGURATIONS(guid'f26bba84-b838-4a14-8598-bf327b026e0a')/CURRENT_ENTRY"
            }
        }
    },
    "SEARCH_FORM": {
        "__metadata": {
            "id": "https://./idmrestapi/v2/service/
TASK_CONFIGURATIONS(guid'a5bdd3e3-c2c8-4daa-98b6-23e9d8b8d387')",
            "uri": "https://./idmrestapi/v2/service/
TASK_CONFIGURATIONS(guid'a5bdd3e3-c2c8-4daa-98b6-23e9d8b8d387')",
            "type": "IdmRestModel.TASK_CONFIGURATIONS"
        },
        "TASK_GUID": "a5bdd3e3-c2c8-4daa-98b6-23e9d8b8d387",
        "TASK_ID": 473,
        "TASK_GROUP": null,
        "TASK_DISPLAY_NAME": "Search for Entry",
        "TASK_HEADER": null,
        "TASK_FOOTER": null,
        "TASK_HAS_SUBMIT_BUTTON": false,
        "TASK_SUBMIT_BUTTON_CAPTION": null,
        "TASK_HAS_MODIFY_BUTTON": false,
        "TASK_MODIFY_BUTTON_CAPTION": null,
        "TASK_CONFIRMATION_MESSAGE": null,
        "TASK_ATTRIBUTES": {
            "__deferred": {
                "uri": "https://./idmrestapi/v2/service/
TASK_CONFIGURATIONS(guid'a5bdd3e3-c2c8-4daa-98b6-23e9d8b8d387')/TASK_ATTRIBUTES"
            }
        },
        "CURRENT_ENTRY": {
            "__deferred": {
                "uri": "https://./idmrestapi/v2/service/
TASK_CONFIGURATIONS(guid'a5bdd3e3-c2c8-4daa-98b6-23e9d8b8d387')/CURRENT_ENTRY"
            }
        }
    },
    "LIST_ATTRIBUTES": {
        "__deferred": {
            "uri": "https://./idmrestapi/v2/service/ENTRY_TYPES('MX_PERSON')/
LIST_ATTRIBUTES"
        }
    }
}
}
}

```

Response Example 3

```

{
  "d": {

```



```

    "__metadata": {
      "id": "https://.idmrestapi/v2/service/ENTRY_TYPES('PROJECT')",
      "uri": "https://.idmrestapi/v2/service/ENTRY_TYPES('PROJECT')",
      "type": "IdmRestModel.ENTRY_TYPES"
    },
    "NAME": "PROJECT",
    "DISPLAY_NAME": "Project",
    "IS_CONTEXT_TYPE": true,
    "IS_SEARCHABLE": true,
    "DISPLAY_FORM": {
      "__metadata": {
        "id": "https://.idmrestapi/v2/service/
TASK_CONFIGURATIONS(guid'c9503582-d438-41c9-99da-b281fac82a3e')",
        "uri": "https://.idmrestapi/v2/service/
TASK_CONFIGURATIONS(guid'c9503582-d438-41c9-99da-b281fac82a3e')",
        "type": "IdmRestModel.TASK_CONFIGURATIONS"
      },
      "TASK_GUID": "c9503582-d438-41c9-99da-b281fac82a3e",
      "TASK_ID": 4399,
      "TASK_GROUP": null,
      "TASK_DISPLAY_NAME": "Project_Display_Form",
      "TASK_HEADER": null,
      "TASK_FOOTER": null,
      "TASK_HAS_SUBMIT_BUTTON": true,
      "TASK_SUBMIT_BUTTON_CAPTION": null,
      "TASK_HAS_MODIFY_BUTTON": true,
      "TASK_MODIFY_BUTTON_CAPTION": null,
      "TASK_CONFIRMATION_MESSAGE": null,
      "TASK_ATTRIBUTES": {
        "results": [
          {
            "__metadata": {
              "id": "https://.idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'c9503582-
d438-41c9-99da-b281fac82a3e',ATTRIBUTE_NAME='--Tab--')",
              "uri": "https://.idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'c9503582-
d438-41c9-99da-b281fac82a3e',ATTRIBUTE_NAME='--Tab--')",
              "type": "IdmRestModel.TaskAttribute"
            },
            "TASK_GUID": "c9503582-d438-41c9-99da-b281fac82a3e",
            "ATTRIBUTE_NAME": "--Tab--",
            "DISPLAY_NAME": "General",
            "DEFAULT_VALUE": null,
            "HIDE_INPUT": false,
            "TOOLTIP": null,
            "VALUEHELP": false,
            "NUMBER_OF_LINES": 0,
            "READONLY": false,
            "MANDATORY": false,
            "IS_MIXED_CASE": false,
            "IS_MIXED_LETTERS_NUMBERS": false,
            "IS_UNIQUE": false,
            "MIN_LENGTH": 0,
            "MAX_LENGTH": 0,
            "VALIDATION_REGEX": "",
            "CONFIRM_INPUT": false,
            "LIST_ENTRIES_ON_LOAD": false,
            "SHOW_SEARCH_FIELD": false,
            "PRESENTATION_TYPE": -3,
            "STORAGE_TYPE": 0,
            "IS_MULTIVALUE": false,
            "REFERENCED_ENTRYTYPE": null,
            "REFERENCED_ENTRYTYPE_DISPLAY_NAME": null,
            "ALLOWED_VALUES": {
              "__deferred": {

```

```

"uri": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'c9503582-
d438-41c9-99da-b281fac82a3e',ATTRIBUTE_NAME='--Tab--')/ALLOWED_VALUES"
    }
  },
  {
    "__metadata": {
      "id": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'c9503582-
d438-41c9-99da-b281fac82a3e',ATTRIBUTE_NAME='DISPLAYNAME')",
      "uri": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'c9503582-
d438-41c9-99da-b281fac82a3e',ATTRIBUTE_NAME='DISPLAYNAME')",
      "type": "IdMRestModel.TaskAttribute"
    },
    "TASK_GUID": "c9503582-d438-41c9-99da-b281fac82a3e",
    "ATTRIBUTE_NAME": "DISPLAYNAME",
    "DISPLAY_NAME": "Display Name",
    "DEFAULT_VALUE": null,
    "HIDE_INPUT": false,
    "TOOLTIP": "",
    "VALUEHELP": false,
    "NUMBER_OF_LINES": 5,
    "READONLY": false,
    "MANDATORY": true,
    "IS_MIXED_CASE": false,
    "IS_MIXED_LETTERS_NUMBERS": false,
    "IS_UNIQUE": false,
    "MIN_LENGTH": 0,
    "MAX_LENGTH": 0,
    "VALIDATION_REGEX": "",
    "CONFIRM_INPUT": false,
    "LIST_ENTRIES_ON_LOAD": true,
    "SHOW_SEARCH_FIELD": false,
    "PRESENTATION_TYPE": 1,
    "STORAGE_TYPE": 1,
    "IS_MULTIVALUE": false,
    "REFERENCED_ENTRYTYPE": null,
    "REFERENCED_ENTRYTYPE_DISPLAY_NAME": null,
    "ALLOWED_VALUES": {
      "__deferred": {
        "uri": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'c9503582-
d438-41c9-99da-b281fac82a3e',ATTRIBUTE_NAME='DISPLAYNAME')/ALLOWED_VALUES"
      }
    }
  },
  {
    "__metadata": {
      "id": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'c9503582-
d438-41c9-99da-b281fac82a3e',ATTRIBUTE_NAME='MSKEYVALUE')",
      "uri": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'c9503582-
d438-41c9-99da-b281fac82a3e',ATTRIBUTE_NAME='MSKEYVALUE')",
      "type": "IdMRestModel.TaskAttribute"
    },
    "TASK_GUID": "c9503582-d438-41c9-99da-b281fac82a3e",
    "ATTRIBUTE_NAME": "MSKEYVALUE",
    "DISPLAY_NAME": "Unique ID",
    "DEFAULT_VALUE": null,
    "HIDE_INPUT": false,
    "TOOLTIP": "",
    "VALUEHELP": false,
    "NUMBER_OF_LINES": 5,
    "READONLY": false,
    "MANDATORY": true,

```

```

        "IS_MIXED_CASE": false,
        "IS_MIXED_LETTERS_NUMBERS": false,
        "IS_UNIQUE": true,
        "MIN_LENGTH": 0,
        "MAX_LENGTH": 0,
        "VALIDATION_REGEX": "",
        "CONFIRM_INPUT": false,
        "LIST_ENTRIES_ON_LOAD": true,
        "SHOW_SEARCH_FIELD": false,
        "PRESENTATION_TYPE": 1,
        "STORAGE_TYPE": 1,
        "IS_MULTIVALUE": false,
        "REFERENCED_ENTRYTYPE": null,
        "REFERENCED_ENTRYTYPE_DISPLAY_NAME": null,
        "ALLOWED_VALUES": {
            "__deferred": {
                "uri": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'c9503582-d438-41c9-99da-b281fac82a3e',ATTRIBUTE_NAME='MSKEYVALUE')/ALLOWED_VALUES"
            }
        }
    },
    ],
    },
    "CURRENT_ENTRY": {
        "__deferred": {
            "uri": "https://./idmrestapi/v2/service/TASK_CONFIGURATIONS(guid'c9503582-d438-41c9-99da-b281fac82a3e')/CURRENT_ENTRY"
        }
    },
    "SEARCH_FORM": {
        "__deferred": {
            "uri": "https://./idmrestapi/v2/service/ENTRY_TYPES('PROJECT')/SEARCH_FORM"
        }
    },
    "LIST_ATTRIBUTES": {
        "__deferred": {
            "uri": "https://./idmrestapi/v2/service/ENTRY_TYPES('PROJECT')/LIST_ATTRIBUTES"
        }
    }
}

```

8.14 Task Access URIs

The URIs described in the sections below allow reading Identity Management Tasks that are defined as user interface tasks (UI). You can search for tasks, read details for a certain task or read task attributes. The query only returns those tasks to which the user has access.

8.14.1 Get All Self-Service Tasks

Returns a collection of all available self-service Tasks.

Self-service tasks are user interface tasks that users can execute for their own entry, for example, to change their personal data. These tasks operate on the entry type of the logged-in user. All logged-in users who are

assigned the rest_v2 UME action have access to this task and can execute it for their own entry only, typically MX_PERSON.

URI

```
(* [GET] *)
```

```
URI = './idmrestapi/v2/service/SELFSERVICETASKS';
```

Request Parameters

None

Request Example

```
GET /idmrestapi/v2/service/SELFSERVICETASKS HTTPS/1.1
```

Response Types

```
Content-Type: application/json | application/atom+xml
```

HTTPS Response Codes

- 200 OK
Response Body: Collection(Task)

Response Example

```
{
  "d": {
    "results": [
      {
        "__metadata": {
          "id": "https://./idmrestapi/v2/service/
SELFSERVICETASKS(guid'd9aa384a-c5d2-4008-adb7-6bcd63ddae72')",
```

```

        "uri": "https://./idmrestapi/v2/service/  

SELFSERVICETASKS(guid'd9aa384a-c5d2-4008-adb7-6bcd63ddae72')",  

        "type": "IdMRestModel.SELFSERVICETASKS"  

    },  

    "TASK_GUID": "d9aa384a-c5d2-4008-adb7-6bcd63ddae72",  

    "TASK_ID": 492,  

    "TASK_GROUP": "Identity",  

    "TASK_DISPLAY_NAME": "Change Own Data",  

    "TASK_HEADER": null,  

    "TASK_FOOTER": "Task to change your data",  

    "TASK_HAS_SUBMIT_BUTTON": true,  

    "TASK_SUBMIT_BUTTON_CAPTION": null,  

    "TASK_HAS_MODIFY_BUTTON": true,  

    "TASK_MODIFY_BUTTON_CAPTION": null,  

    "TASK_CONFIRMATION_MESSAGE": "Identity changed",  

    "TASK_ATTRIBUTES": {  

        "__deferred": {  

            "uri": "https://./idmrestapi/v2/service/  

SELFSERVICETASKS(guid'd9aa384a-c5d2-4008-adb7-6bcd63ddae72')/TASK_ATTRIBUTES"  

        }  

    },  

    "CURRENT_ENTRY": {  

        "__deferred": {  

            "uri": "https://./idmrestapi/v2/service/  

SELFSERVICETASKS(guid'd9aa384a-c5d2-4008-adb7-6bcd63ddae72')/CURRENT_ENTRY"  

        }  

    }  

},  

{  

    "__metadata": {  

        "id": "https://./idmrestapi/v2/service/  

SELFSERVICETASKS(guid'860e8b87-59f6-4fa6-a23d-b444d05816e4')",  

        "uri": "https://./idmrestapi/v2/service/  

SELFSERVICETASKS(guid'860e8b87-59f6-4fa6-a23d-b444d05816e4')",  

        "type": "IdMRestModel.SELFSERVICETASKS"  

    },  

    "TASK_GUID": "860e8b87-59f6-4fa6-a23d-b444d05816e4",  

    "TASK_ID": 500,  

    "TASK_GROUP": "Identity",  

    "TASK_DISPLAY_NAME": "Request New Role",  

    "TASK_HEADER": null,  

    "TASK_FOOTER": "Task for requesting new roles",  

    "TASK_HAS_SUBMIT_BUTTON": true,  

    "TASK_SUBMIT_BUTTON_CAPTION": "Submit",  

    "TASK_HAS_MODIFY_BUTTON": true,  

    "TASK_MODIFY_BUTTON_CAPTION": null,  

    "TASK_CONFIRMATION_MESSAGE": "Request submitted",  

    "TASK_ATTRIBUTES": {  

        "__deferred": {  

            "uri": "https://./idmrestapi/v2/service/  

SELFSERVICETASKS(guid'860e8b87-59f6-4fa6-a23d-b444d05816e4')/TASK_ATTRIBUTES"  

        }  

    },  

    "CURRENT_ENTRY": {  

        "__deferred": {  

            "uri": "https://./idmrestapi/v2/service/  

SELFSERVICETASKS(guid'860e8b87-59f6-4fa6-a23d-b444d05816e4')/CURRENT_ENTRY"  

        }  

    }  

}  

}]  

}

```

8.14.2 Get a Specific Self-Service Task

Returns only one specific self-service task indicated by its task GUID.

If the provided task GUID is not from a self-service task, the call returns an error response.

Because of the special semantics of these tasks, the entry access is limited to the currently logged-in user. Therefore, the returned task entity contains the additional dynamic property "CURRENT_ENTRY" with the entry access URI for the current user and this task, as specified in *Get Specific Entry* under Response Examples.

URI

```
(* [GET] *)  
URI      = './idmrestapi/v2/service/SELFSERVICETASKS('TaskGUID')';  
TaskGUID = Edm.Guid;
```

Request Parameters

Supported Query Options:

- \$expand: To inline deferred current entry or task attributes.

Request Example 1

```
GET /idmrestapi/v2/service/SELFSERVICETASKS(guid'494E68D9-ECCE-41F5-AE3D-B93D23F54E70') HTTPS/1.1
```

Request Example 2

Get specified self-service task and execute it (get attributes + values) for the current user.

This request returns all attributes of the MX_PERSON entry type and their attribute values if they are defined in the corresponding self-service task. Regardless of whether they are defined in the self-service task, the following attribute values are always returned : ID (MSKEY), SV_MSKEYVALUE, SV_DISPLAYNAME and BI_MX_USER_PICTURE.

```
GET /idmrestapi/v2/service/SELFSERVICETASKS(guid'494E68D9-ECCE-41F5-AE3D-B93D23F54E70')?$expand=CURRENT_ENTRY HTTPS/1.1
```

Response Types

- Content-Type: application/json | application/atom+xml

HTTPS Response Codes

- 200 OK
Response Body: Task
- 403 Forbidden
 - Task GUID is not from a self-service task.
-

Response Example 1

```
{
  "d": {
    "__metadata": {
      "id": "https://./idmrestapi/v2/service/
SELFSEVICETASKS(guid'd9aa384a-c5d2-4008-adb7-6bcd63ddae72')",
      "uri": "https://./idmrestapi/v2/service/
SELFSEVICETASKS(guid'd9aa384a-c5d2-4008-adb7-6bcd63ddae72')",
      "type": "IdMRestModel.SELFSEVICETASKS"
    },
    "TASK_GUID": "d9aa384a-c5d2-4008-adb7-6bcd63ddae72",
    "TASK_ID": 492,
    "TASK_GROUP": null,
    "TASK_DISPLAY_NAME": "Change Own Data",
    "TASK_HEADER": null,
    "TASK_FOOTER": "Task to change your data",
    "TASK_HAS_SUBMIT_BUTTON": true,
    "TASK_SUBMIT_BUTTON_CAPTION": null,
    "TASK_HAS_MODIFY_BUTTON": true,
    "TASK_MODIFY_BUTTON_CAPTION": null,
    "TASK_CONFIRMATION_MESSAGE": "Identity changed",
    "TASK_ATTRIBUTES": {
      "__deferred": {
        "uri": "https://./idmrestapi/v2/service/
SELFSEVICETASKS(guid'd9aa384a-c5d2-4008-adb7-6bcd63ddae72')/TASK_ATTRIBUTES"
      }
    },
    "CURRENT_ENTRY": {
      "__deferred": {
        "uri": "https://./idmrestapi/v2/service/
SELFSEVICETASKS(guid'd9aa384a-c5d2-4008-adb7-6bcd63ddae72')/CURRENT_ENTRY"
      }
    }
  }
}
```

Response Example 2

```
{
  "d": {
    "__metadata": {
      "id": "https://./idmrestapi/v2/service/
SELFSEVICETASKS(guid'd9aa384a-c5d2-4008-adb7-6bcd63ddae72')",
      "uri": "https://./idmrestapi/v2/service/
SELFSEVICETASKS(guid'd9aa384a-c5d2-4008-adb7-6bcd63ddae72')",
      "type": "IdMRestModel.SELFSEVICETASKS"
    },
    "TASK_GUID": "d9aa384a-c5d2-4008-adb7-6bcd63ddae72",
    "TASK_ID": 492,
    "TASK_GROUP": null,
    "TASK_DISPLAY_NAME": "Change Own Data",
    "TASK_HEADER": null,
    "TASK_FOOTER": "Task to change your data",
    "TASK_HAS_SUBMIT_BUTTON": true,
    "TASK_SUBMIT_BUTTON_CAPTION": null,
    "TASK_HAS_MODIFY_BUTTON": true,
    "TASK_MODIFY_BUTTON_CAPTION": null,
    "TASK_CONFIRMATION_MESSAGE": "Identity changed",
    "TASK_ATTRIBUTES": {
      "__deferred": {
        "uri": "https://./idmrestapi/v2/service/
SELFSEVICETASKS(guid'd9aa384a-c5d2-4008-adb7-6bcd63ddae72')/TASK_ATTRIBUTES"
      }
    },
    "CURRENT_ENTRY": {
      "__metadata": {
        "id": "https://./idmrestapi/v2/service/
ET_MX_PERSON(ID=23,TASK_GUID=guid'd9aa384a-c5d2-4008-adb7-6bcd63ddae72')",
        "uri": "https://./idmrestapi/v2/service/
ET_MX_PERSON(ID=23,TASK_GUID=guid'd9aa384a-c5d2-4008-adb7-6bcd63ddae72')",
        "type": "IdMRestModel.ET_MX_PERSON"
      },
      "ID": 23,
      "TASK_GUID": "d9aa384a-c5d2-4008-adb7-6bcd63ddae72",
      "SV_MSKEYVALUE": "benz1",
      "SV_DISPLAYNAME": "Benz",
      "SV_MX_FIRSTNAME": "Marie",
      "SV_MX_LASTNAME": "Benz",
      "SV_MX_MAIL_PRIMARY": "marie.benz@example.com",
      "BI_MX_USER_PICTURE": {
        "__deferred": {
          "uri": "https://./idmrestapi/v2/
service/ET_MX_PERSON(ID=23,TASK_GUID=guid'7227f322-87a7-423b-a6d0-1dcb6fa9778f')/
BI_MX_USER_PICTURE"
        }
      }
    }
  }
}
```

Related Information

[Get Specific Entry \[page 857\]](#)

8.14.3 Get All Task Attributes

Return information about all the attributes that a task specified by its TaskGUID exposes of an entry type and all layout attributes (that is, Tab, Label, Section, Column, Line). For details about the properties of a task attribute, see *Get a Specific Task Attribute*.

URI

(* [GET] *)

URI = './idmrestapi/v2/service/TASK_ATTRIBUTES';

Request Parameters

Parameter	= '?' Filter
Filter	= TaskGUIDFilter;
TaskGUIDFilter	= 'TASK_GUID=' TaskGUID
TaskGUID	= Edm.Guid;

Note

TASK_GUID is a mandatory filter attribute.

Request Example

```
GET /idmrestapi/v2/service/TASK_ATTRIBUTES?TASK_GUID=guid'35B264BF-A75A-447D-A4EA-7894725245CE'  
HTTPS/1.1
```

Response Types

Content-Type: application/json | application/atom+xml

HTTPS Response Codes

- 200 OK

Response Body: Collection(TaskAttribute)

- 403 Forbidden
Logged-in user does not have the necessary UME actions. See [Authentication and Authorization \[page 829\]](#)
- 404 Not Found
The specified task does not exist.

Response Example

```
{
  "d": {
    "results": [
      {
        "__metadata": {
          "id": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'baa786c2-75f5-4235-b38b-088eae57c09e',ATTRIBUTE_NAME='--Tab--')",
          "uri": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'baa786c2-75f5-4235-b38b-088eae57c09e',ATTRIBUTE_NAME='--Tab--')",
          "type": "IdMRestModel.TaskAttribute"
        },
        "TASK_GUID": "baa786c2-75f5-4235-b38b-088eae57c09e",
        "ATTRIBUTE_NAME": "--Tab--",
        "DISPLAY_NAME": "General",
        "DEFAULT_VALUE": null,
        "HIDE_INPUT": false,
        "TOOLTIP": null,
        "VALUEHELP": false,
        "NUMBER_OF_LINES": 0,
        "READONLY": false,
        "MANDATORY": false,
        "IS_MIXED_CASE": false,
        "IS_MIXED_LETTERS_NUMBERS": false,
        "IS_UNIQUE": false,
        "MIN_LENGTH": 0,
        "MAX_LENGTH": 0,
        "VALIDATION_REGEX": "",
        "CONFIRM_INPUT": false,
        "LIST_ENTRIES_ON_LOAD": false,
        "SHOW_SEARCH_FIELD": false,
        "PRESENTATION_TYPE": -3,
        "STORAGE_TYPE": 0,
        "IS_MULTIVALUE": false,
        "REFERENCED_ENTRYTYPE": null,
        "REFERENCED_ENTRYTYPE_DISPLAY_NAME": null,
        "ALLOWED_VALUES": {
          "__deferred": {
            "uri": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'baa786c2-75f5-4235-b38b-088eae57c09e',ATTRIBUTE_NAME='--Tab--')/ALLOWED_VALUES"
          }
        }
      },
      {
        "__metadata": {
          "id": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'baa786c2-75f5-4235-b38b-088eae57c09e',ATTRIBUTE_NAME='DISPLAYNAME')",
```

```

        "uri": "https://./
idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'baa786c2-75f5-4235-
b38b-088eae57c09e',ATTRIBUTE_NAME='DISPLAYNAME')",
        "type": "IdmRestModel.TaskAttribute"
    },
    "TASK_GUID": "baa786c2-75f5-4235-b38b-088eae57c09e",
    "ATTRIBUTE_NAME": "DISPLAYNAME",
    "DISPLAY_NAME": "Display Name",
    "DEFAULT_VALUE": null,
    "HIDE_INPUT": false,
    "TOOLTIP": "",
    "VALUEHELP": false,
    "NUMBER_OF_LINES": 5,
    "READONLY": false,
    "MANDATORY": false,
    "IS_MIXED_CASE": false,
    "IS_MIXED_LETTERS_NUMBERS": false,
    "IS_UNIQUE": false,
    "MIN_LENGTH": 0,
    "MAX_LENGTH": 0,
    "VALIDATION_REGEX": "",
    "CONFIRM_INPUT": false,
    "LIST_ENTRIES_ON_LOAD": true,
    "SHOW_SEARCH_FIELD": false,
    "PRESENTATION_TYPE": 1,
    "STORAGE_TYPE": 1,
    "IS_MULTIVALUE": false,
    "REFERENCED_ENTRYTYPE": null,
    "REFERENCED_ENTRYTYPE_DISPLAY_NAME": null,
    "ALLOWED_VALUES": {
        "__deferred": {
            "uri": "https://./
idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'baa786c2-75f5-4235-
b38b-088eae57c09e',ATTRIBUTE_NAME='DISPLAYNAME')/ALLOWED_VALUES"
        }
    }
},
{
    "__metadata": {
        "id": "https://./
idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'baa786c2-75f5-4235-
b38b-088eae57c09e',ATTRIBUTE_NAME='MSKEYVALUE')",
        "uri": "https://./
idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'baa786c2-75f5-4235-
b38b-088eae57c09e',ATTRIBUTE_NAME='MSKEYVALUE')",
        "type": "IdmRestModel.TaskAttribute"
    },
    "TASK_GUID": "baa786c2-75f5-4235-b38b-088eae57c09e",
    "ATTRIBUTE_NAME": "MSKEYVALUE",
    "DISPLAY_NAME": "Unique ID",
    "DEFAULT_VALUE": null,
    "HIDE_INPUT": false,
    "TOOLTIP": "",
    "VALUEHELP": false,
    "NUMBER_OF_LINES": 5,
    "READONLY": false,
    "MANDATORY": true,
    "IS_MIXED_CASE": false,
    "IS_MIXED_LETTERS_NUMBERS": false,
    "IS_UNIQUE": true,
    "MIN_LENGTH": 0,
    "MAX_LENGTH": 0,
    "VALIDATION_REGEX": "",
    "CONFIRM_INPUT": false,
    "LIST_ENTRIES_ON_LOAD": true,
    "SHOW_SEARCH_FIELD": false,
    "PRESENTATION_TYPE": 1,
    "STORAGE_TYPE": 1,

```

```

        "IS_MULTIVALUE": false,
        "REFERENCED_ENTRYTYPE": null,
        "REFERENCED_ENTRYTYPE_DISPLAY_NAME": null,
        "ALLOWED_VALUES": {
            "__deferred": {
                "uri": "https://./
idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'baa786c2-75f5-4235-
b38b-088eae57c09e',ATTRIBUTE_NAME='MSKEYVALUE')/ALLOWED_VALUES"
            }
        }
    },
    ],
}

```

Related Information

[Get a Specific Task Attribute \[page 918\]](#)

8.14.4 Get a Specific Task Attribute

Returns only a specific task attribute specified by the TaskGUID and the task attribute name.

The properties of a task attribute are a subset of the properties of an attribute exposed by the schema information requests. Only the properties are contained that can be overwritten on task level. For the complete list, see the metadata.xml file. If a property is not set on the task level, the property value of the attribute in the schema is returned as a fallback.

URI

```

(* [GET] *)
URI                = './idmrestapi/v2/service/
TASK_ATTRIBUTES(ATTRIBUTE_NAME='TaskAttributeName',TASK_GUID=' TaskGUID ');
TaskGUID           = Edm.Guid;
TaskAttributeName  = Edm.String;

```

Request Parameters

Supported Query Options:

- \$expand: To inline deferred allowed values.

Request Example 1

```
GET /idmrestapi/v2/service/  
TASK_ATTRIBUTES(ATTRIBUTE_NAME='MX_FIRSTNAME',TASK_GUID=guid'35B264BF-A75A-447D-  
A4EA-7894725245CE') HTTP/1.1
```

Request Example 2

```
GET /idmrestapi/v2/service/  
TASK_ATTRIBUTES(ATTRIBUTE_NAME='MX_SALUTATION',TASK_GUID=guid'35B264BF-A75A-447D-  
A4EA-7894725245CE')?$expand=ALLOWED_VALUES HTTP/1.1
```

Response Types

```
Content-Type: application/json | application/atom+xml
```

HTTPS Response Codes

- 200 OK
Response body: TaskAttribute
- 403 Forbidden
Logged-in user is not allowed to access the task.
- 404 Not Found
The specified task does not exist.

Response Example 1

```
{  
  "d": {  
    "__metadata": {  
      "id": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'd9aa384a-  
c5d2-4008-adb7-6bcd63ddae72',ATTRIBUTE_NAME='MX_FIRSTNAME')",  
      "uri": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'd9aa384a-  
c5d2-4008-adb7-6bcd63ddae72',ATTRIBUTE_NAME='MX_FIRSTNAME')",  
      "type": "IdMRestModel.TaskAttribute"  
    },  
    "TASK_GUID": "d9aa384a-c5d2-4008-adb7-6bcd63ddae72",  
    "ATTRIBUTE_NAME": "MX_FIRSTNAME",  
    "DISPLAY_NAME": "First Name",  
    "DEFAULT_VALUE": null,  
  }  
}
```

```

        "HIDE_INPUT": false,
        "TOOLTIP": "",
        "VALUEHELP": false,
        "NUMBER_OF_LINES": 5,
        "READONLY": true,
        "MANDATORY": false,
        "IS_MIXED_CASE": false,
        "IS_MIXED_LETTERS_NUMBERS": false,
        "IS_UNIQUE": false,
        "MIN_LENGTH": 0,
        "MAX_LENGTH": 0,
        "VALIDATION_REGEX": "",
        "CONFIRM_INPUT": false,
        "LIST_ENTRIES_ON_LOAD": true,
        "SHOW_SEARCH_FIELD": false,
        "PRESENTATION_TYPE": 1,
        "STORAGE_TYPE": 1,
        "IS_MULTIVALUE": false,
        "REFERENCED_ENTRYTYPE": null,
        "REFERENCED_ENTRYTYPE_DISPLAY_NAME": null,
        "ALLOWED_VALUES": {
            "__deferred": {
                "uri": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'd9aa384a-
c5d2-4008-adb7-6bcd63ddae72',ATTRIBUTE_NAME='MX_FIRSTNAME')/ALLOWED_VALUES"
            }
        }
    }
}

```

Response Example 2 (\$expand=ALLOWED_VALUES)

```

{
    "d": {
        "__metadata": {
            "id": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'd9aa384a-
c5d2-4008-adb7-6bcd63ddae72',ATTRIBUTE_NAME='MX_SALUTATION')",
            "uri": "https://./idmrestapi/v2/service/TASK_ATTRIBUTES(TASK_GUID=guid'd9aa384a-
c5d2-4008-adb7-6bcd63ddae72',ATTRIBUTE_NAME='MX_SALUTATION')",
            "type": "IdmRestModel.TaskAttribute"
        },
        "TASK_GUID": "d9aa384a-c5d2-4008-adb7-6bcd63ddae72",
        "ATTRIBUTE_NAME": "MX_SALUTATION",
        "DISPLAY_NAME": "Title",
        "DEFAULT_VALUE": null,
        "HIDE_INPUT": false,
        "TOOLTIP": "",
        "VALUEHELP": false,
        "NUMBER_OF_LINES": 5,
        "READONLY": true,
        "MANDATORY": false,
        "IS_MIXED_CASE": false,
        "IS_MIXED_LETTERS_NUMBERS": false,
        "IS_UNIQUE": false,
        "MIN_LENGTH": 0,
        "MAX_LENGTH": 0,
        "VALIDATION_REGEX": "",
        "CONFIRM_INPUT": false,
        "LIST_ENTRIES_ON_LOAD": true,
        "SHOW_SEARCH_FIELD": false,
        "PRESENTATION_TYPE": 3,
    }
}

```

```

"STORAGE_TYPE": 1,
"IS_MULTIVALUE": false,
"REFERENCED_ENTRYTYPE": null,
"REFERENCED_ENTRYTYPE_DISPLAY_NAME": null,
"ALLOWED_VALUES": {
  "results": [
    {
      "__metadata": {
        "id": "https://.idmrestapi/v2/service/
ALLOWED_VALUES(ATTRIBUTE_NAME='MX_SALUTATION', TASK_GUID=guid'd9aa384a-c5d2-4008-
adb7-6bcd63ddae72', VALUE_KEY='')",
        "uri": "https://.idmrestapi/v2/service/
ALLOWED_VALUES(ATTRIBUTE_NAME='MX_SALUTATION', TASK_GUID=guid'd9aa384a-c5d2-4008-
adb7-6bcd63ddae72', VALUE_KEY='')",
        "type": "IdMRestModel.AllowedValue"
      },
      "ATTRIBUTE_NAME": "MX_SALUTATION",
      "TASK_GUID": "d9aa384a-c5d2-4008-adb7-6bcd63ddae72",
      "VALUE_KEY": "",
      "VALUE_TEXT": ""
    },
    {
      "__metadata": {
        "id": "https://.idmrestapi/v2/service/
ALLOWED_VALUES(ATTRIBUTE_NAME='MX_SALUTATION', TASK_GUID=guid'd9aa384a-c5d2-4008-
adb7-6bcd63ddae72', VALUE_KEY='Mr.')",
        "uri": "https://.idmrestapi/v2/service/
ALLOWED_VALUES(ATTRIBUTE_NAME='MX_SALUTATION', TASK_GUID=guid'd9aa384a-c5d2-4008-
adb7-6bcd63ddae72', VALUE_KEY='Mr.')",
        "type": "IdMRestModel.AllowedValue"
      },
      "ATTRIBUTE_NAME": "MX_SALUTATION",
      "TASK_GUID": "d9aa384a-c5d2-4008-adb7-6bcd63ddae72",
      "VALUE_KEY": "Mr.",
      "VALUE_TEXT": "Mr."
    },
    {
      "__metadata": {
        "id": "https://.idmrestapi/v2/service/
ALLOWED_VALUES(ATTRIBUTE_NAME='MX_SALUTATION', TASK_GUID=guid'd9aa384a-c5d2-4008-
adb7-6bcd63ddae72', VALUE_KEY='Ms.')",
        "uri": "https://.idmrestapi/v2/service/
ALLOWED_VALUES(ATTRIBUTE_NAME='MX_SALUTATION', TASK_GUID=guid'd9aa384a-c5d2-4008-
adb7-6bcd63ddae72', VALUE_KEY='Ms.')",
        "type": "IdMRestModel.AllowedValue"
      },
      "ATTRIBUTE_NAME": "MX_SALUTATION",
      "TASK_GUID": "d9aa384a-c5d2-4008-adb7-6bcd63ddae72",
      "VALUE_KEY": "Ms.",
      "VALUE_TEXT": "Ms."
    }
  ]
}
}
}
}

```

8.14.5 Get Allowed Values of a Specific Task Attribute

This access gets a list of the permissible values for a certain task attribute.

The property "ALLOWED_VALUES" specifies the values that are allowed for a specific attribute. It returns the evaluated list of all allowed values dependent on the type of the allowed (legal) values:

- Values: The specified list of values is returned.
- SQL Query: The result list of the query is returned.
- Value help: Returns the list of all keys with their descriptions from the value help table for this attribute.
- Task: The result of the task is returned.

If no allowed values are defined for the attribute on the UI task level, the allowed values defined on the attribute in the identity store schema are returned.

Note

If the attribute is of data type entry reference, the number of returned allowed values is affected by the [Max search result size](#) setting. You can find more information in [Search Entries \[page 851\]](#).

The value help can be language-dependent, for example, salutation, title, and job title. The values returned are returned in the language of the currently logged-in user. For more information about how the language is determined, see also *Localization*.

URI

```
(* [GET] *)  
URI      = './idmrestapi/v2/service/ALLOWED_VALUES'
```

Request Parameters

```
Parameter      = '?' Filter  
Filter          = '$filter=' MandatoryFilter [Junction OptionalFilter];  
MandatoryFilter = TaskGUIDFilter Junction TaskAttributeFilter;  
OptionalFilter  = KeyValuePair [Junction KeyValuePair];  
TaskGUIDFilter  = 'TASK_GUID' Comparator TaskGUID  
TaskAttributeFilter = 'ATTRIBUTE_NAME' Comparator AttributeName  
KeyValuePair     = FilterPropertyName Comparator SearchValue;  
FilterPropertyName = 'VALUE_KEY' | 'VALUE_TEXT'  
UserMskeyFilter  = 'USERMSKEY' Comparator UserMskey  
Junction         = 'and';  
Comparator       = 'eq';  
TaskGUID         = Edm.Guid;  
AttributeName    = "" Edm.String "";  
SearchValue      = "" Edm.String "" (* * is a wildcard *)  
UserMskey        = "" Edm.String "";
```


Note

- The filter parameter is case sensitive, except for the search value.
- TASK_GUID and ATTRIBUTE_NAME are mandatory filter attributes.
- The URI has to be URL escaped to replace special characters, like the space with "%20".
- The filter properties VALUE_KEY and VALUE_TEXT work only when the allowed values of type value help are returned or if the attribute is of data type entry reference.
- If the allowed values type is Values, then allowed values is added with empty VALUE_KEY and VALUE_TEXT to the list of permissible values.

Request Example1

```
[GET]/idmrestapi/v2/service/ALLOWED_VALUES?$filter=ATTRIBUTE_NAME eq  
'MX_SALUTATION' and TASK_GUID eq guid'35B264BF-A75A-447D-A4EA-7894725245CE'
```

Request Example2

Request the values of entry reference attributes.

```
[GET]/idmrestapi/v2/service/ALLOWED_VALUES?$filter=ATTRIBUTE_NAME eq  
'MX_MANAGER' and TASK_GUID eq guid'D9AA384A-C5D2-4008-ADB7-6BCD63DDAE72'
```

Request Example3

Request the values of entry reference attributes that have no reference entry type defined for them. In this example, request only the roles of MX_ASSIGNMENT entry reference attribute.

```
[GET]/idmrestapi/v2/service/ALLOWED_VALUES?$filter=ATTRIBUTE_NAME eq  
'MX_ASSIGNMENT' and TASK_GUID eq guid'A56DB5FE-DA5A-4109-8518-054B7E0FA454'  
&ENTRY_TYPE=MX_ROLE
```

Response Example1

```
{  
  "d": {  
    "results": [  
      {  
        "__metadata": {  
          "id": "https://./idmrestapi/v2/service/  
ALLOWED_VALUES(ATTRIBUTE_NAME='MX_SALUTATION', TASK_GUID=guid'd9aa384a-c5d2-4008-  
adb7-6bcd63ddae72', VALUE_KEY='')",
```

```

        "uri": "https://./idmrestapi/v2/service/
ALLOWED_VALUES(ATTRIBUTE_NAME='MX_SALUTATION', TASK_GUID=guid'd9aa384a-c5d2-4008-
adb7-6bcd63ddae72', VALUE_KEY='')",
        "type": "IdMRestModel.AllowedValue"
    },
    "ATTRIBUTE_NAME": "MX_SALUTATION",
    "TASK_GUID": "d9aa384a-c5d2-4008-adb7-6bcd63ddae72",
    "VALUE_KEY": "",
    "VALUE_TEXT": ""
},
{
    "__metadata": {
        "id": "https://./idmrestapi/v2/service/
ALLOWED_VALUES(ATTRIBUTE_NAME='MX_SALUTATION', TASK_GUID=guid'd9aa384a-c5d2-4008-
adb7-6bcd63ddae72', VALUE_KEY='Mr.')",
        "uri": "https://./idmrestapi/v2/service/
ALLOWED_VALUES(ATTRIBUTE_NAME='MX_SALUTATION', TASK_GUID=guid'd9aa384a-c5d2-4008-
adb7-6bcd63ddae72', VALUE_KEY='Mr.')",
        "type": "IdMRestModel.AllowedValue"
    },
    "ATTRIBUTE_NAME": "MX_SALUTATION",
    "TASK_GUID": "d9aa384a-c5d2-4008-adb7-6bcd63ddae72",
    "VALUE_KEY": "Mr.",
    "VALUE_TEXT": "Mr."
},
{
    "__metadata": {
        "id": "https://./idmrestapi/v2/service/
ALLOWED_VALUES(ATTRIBUTE_NAME='MX_SALUTATION', TASK_GUID=guid'd9aa384a-c5d2-4008-
adb7-6bcd63ddae72', VALUE_KEY='Ms.')",
        "uri": "https://./idmrestapi/v2/service/
ALLOWED_VALUES(ATTRIBUTE_NAME='MX_SALUTATION', TASK_GUID=guid'd9aa384a-c5d2-4008-
adb7-6bcd63ddae72', VALUE_KEY='Ms.')",
        "type": "IdMRestModel.AllowedValue"
    },
    "ATTRIBUTE_NAME": "MX_SALUTATION",
    "TASK_GUID": "d9aa384a-c5d2-4008-adb7-6bcd63ddae72",
    "VALUE_KEY": "Ms.",
    "VALUE_TEXT": "Ms."
}
]
}
}

```

Response Example2

Return the values of entry reference attributes.

```

{
    "d": {
        "results": [
            {
                "__metadata": {
                    "id": "https://./idmrestapi/v2/service/
ALLOWED_VALUES(ATTRIBUTE_NAME='MX_MANAGER', TASK_GUID=guid'd9aa384a-c5d2-4008-
adb7-6bcd63ddae72', VALUE_KEY='23')",
                    "uri": "https://./idmrestapi/v2/service/
ALLOWED_VALUES(ATTRIBUTE_NAME='MX_MANAGER', TASK_GUID=guid'd9aa384a-c5d2-4008-
adb7-6bcd63ddae72', VALUE_KEY='23')",
                    "type": "IdMRestModel.AllowedValue"
                },
                "ATTRIBUTE_NAME": "MX_MANAGER",
                "TASK_GUID": "d9aa384a-c5d2-4008-adb7-6bcd63ddae72",

```

```

        "VALUE_KEY": "23",
        "VALUE_TEXT": "benz1"
    },
    ]
}

```

Response Example3

```

{
  "d": {
    "results": [
      {
        "__metadata": {
          "id": "https://.idmrestapi/v2/service/ALLOWED_VALUES(ATTRIBUTE_NAME='MX_ASSIGNMENT', TASK_GUID=guid'a56db5fe-da5a-4109-8518-054b7e0fa454', VALUE_KEY='59436')",
          "uri": "https://.idmrestapi/v2/service/ALLOWED_VALUES(ATTRIBUTE_NAME='MX_ASSIGNMENT', TASK_GUID=guid'a56db5fe-da5a-4109-8518-054b7e0fa454', VALUE_KEY='59436')",
          "type": "IdMRestModel.AllowedValue"
        },
        "ATTRIBUTE_NAME": "MX_ASSIGNMENT",
        "TASK_GUID": "a56db5fe-da5a-4109-8518-054b7e0fa454",
        "VALUE_KEY": "59436",
        "VALUE_TEXT": "auditRole",
        "USERSKEY": "25"
      }
    ],
    "__metadata": {
      "id": "https://.idmrestapi/v2/service/ALLOWED_VALUES(ATTRIBUTE_NAME='MX_ASSIGNMENT', TASK_GUID=guid'a56db5fe-da5a-4109-8518-054b7e0fa454', VALUE_KEY='497')",
      "uri": "https://.idmrestapi/v2/service/ALLOWED_VALUES(ATTRIBUTE_NAME='MX_ASSIGNMENT', TASK_GUID=guid'a56db5fe-da5a-4109-8518-054b7e0fa454', VALUE_KEY='497')",
      "type": "IdMRestModel.AllowedValue"
    },
    "ATTRIBUTE_NAME": "MX_ASSIGNMENT",
    "TASK_GUID": "a56db5fe-da5a-4109-8518-054b7e0fa454",
    "VALUE_KEY": "497",
    "VALUE_TEXT": "CONTEXT_ROLE",
    "USERSKEY": "25"
  }
},
    {
      "__metadata": {
        "id": "https://.idmrestapi/v2/service/ALLOWED_VALUES(ATTRIBUTE_NAME='MX_ASSIGNMENT', TASK_GUID=guid'a56db5fe-da5a-4109-8518-054b7e0fa454', VALUE_KEY='24')",
        "uri": "https://.idmrestapi/v2/service/ALLOWED_VALUES(ATTRIBUTE_NAME='MX_ASSIGNMENT', TASK_GUID=guid'a56db5fe-da5a-4109-8518-054b7e0fa454', VALUE_KEY='24')",
        "type": "IdMRestModel.AllowedValue"
      },
      "ATTRIBUTE_NAME": "MX_ASSIGNMENT",
      "TASK_GUID": "a56db5fe-da5a-4109-8518-054b7e0fa454",
      "VALUE_KEY": "24",
      "VALUE_TEXT": "MX_ROLE:ADMIN",
      "USERSKEY": "25"
    }
  ]
}

```

8.15 Approval Access URIs

You can use the following URIs to access SAP Identity Management approval items, for example, changes to entries that do not take effect immediately, but need the approval of an authorized person. You can query for approvals that the currently logged-in user can process, by accessing these items and approving or declining them. Some of the general URIs can also allow you to access the attestation items.

Related Information

[Attestation Access URIs \[page 937\]](#)

8.15.1 Get Open Approval Requests for Current User

This GET request returns a list containing approval items assigned to the logged-in user. The list includes all of the information about the approval item that is needed for an approver to decide if the item can be approved. The list can also contain information about the attestation task.

URI

```
(* [GET] *)  
URI = './idmrestapi/v2/service/TaskCollection'
```

Request Parameters

None.

Request Example

```
[GET]/idmrestapi/v2/service/TaskCollection
```

Response Types

Content-Type: application/json

HTTPS Response Codes

- 200 - OK
- 401 - Unauthorized
If the user is not authenticated.
- 403 - Forbidden
If the user is not authorized to query for open approvals.
- 405 - Method Not Allowed
If the URI was not called with a GET request.

Response Example

```
{
  "d": {
    "results": [
      {
        "__metadata": {
          "uri": "/idmrestapi/v2/service/
TaskCollection(InstanceID='1x7', SAP__Origin='IDM')",
          "type": "TASKPROCESSING.Task"
        },
        "SAP__Origin": "IDM",
        "InstanceID": "1x7",
        "TaskDefinitionID": "1",
        "CreatedBy": "Maximilian Muster",
        "CreatedOn": "\\Date(1362355200000)\\",
        "ExpiryDate": "\\Date(1364947200000)\\",
        "CREATED_BY_ID": 318,
        "REQUESTED_ENTRY_ID": 218,
        "REQUESTED_ENTRY_NAME": "SAP_SDIC_SUPER_ADMINISTRATOR-Role (ASJAVA730)",
        "VALID_FROM": "\\Date(1362355200000)\\",
        "VALID_TO": "\\Date(1364256000000)\\",
        "REASON": "Need SDIC Super Admin rights",
        "CONTEXT": null,
        "TaskTitle": "Approval Task",
        "TaskName": "Approve Users Roles",
        "OPERATION_ID": 1,
        "OPERATION_NAME": "Add",
        "REQUESTED_BY_ID": -3,
        "REQUESTED_BY_NAME": null,
        "TASK_GUID": "F310D68D-C76C-4FE0-A7D2-D0431F390176",

        "CustomAttributeData": {
          "__deferred": {
            "uri": "/idmrestapi/v2/service/
TaskCollection(InstanceID='1x7', SAP__Origin='IDM')/CustomAttributeData"
          }
        },
        "TaskDefinitionData": {
```

```

    "__deferred":{
      "uri":"/idmrestapi/v2/service/
TaskCollection(InstanceID='1x7',SAP__Origin='IDM')/TaskDefinitionData"
    },
    "REFERENCED_ENTRY":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(InstanceID='1x7',SAP__Origin='IDM')/REFERENCED_ENTRY"
      }
    },
    "CONTEXT_ENTRY":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(InstanceID='1x7',SAP__Origin='IDM')/CONTEXT_ENTRY"
      }
    },
    {
      "__metadata":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(InstanceID='1x3',SAP__Origin='IDM')",
        "type":"TASKPROCESSING.Task"
      },
      "SAP__Origin":"IDM",
      "InstanceID":"1x3",
      "TaskDefinitionID":"1",
      "CreatedBy":"Hans Müller",
      "CreatedOn":"\\Date(1362009600000)\\",
      "ExpiryDate":"\\Date(1364601600000)\\",
      "CREATED_BY_ID":318,
      "REQUESTED_ENTRY_ID":195,
      "REQUESTED_ENTRY_NAME":"Administrators-Group (ASJAVA730)",
      "VALID_FROM":null,
      "VALID_TO":"\\Date(1363824000000)\\",
      "REASON":null,
      "CONTEXT": {
        "ID": 2,
        "DISPLAY_NAME":"ProjectX",
        "ENTRYTYPE_DISPLAY_NAME": "Project "
      },
      "OPERATION_ID": 1,
      "OPERATION_NAME": "Add",
      "REQUESTED_BY_ID": -3,
      "REQUESTED_BY_NAME": null,
      "CustomAttributeData":{
        "__deferred":{
          "uri":"/idmrestapi/v2/service/
TaskCollection(InstanceID='1x3',SAP__Origin='IDM')/CustomAttributeData"
        }
      },
      "TaskDefinitionData":{
        "__deferred":{
          "uri":"/idmrestapi/v2/service/
TaskCollection(InstanceID='1x3',SAP__Origin='IDM')/TaskDefinitionData"
        }
      },
      "REFERENCED_ENTRY":{
        "__deferred":{
          "uri":"/idmrestapi/v2/service/
TaskCollection(InstanceID='1x3',SAP__Origin='IDM')/REFERENCED_ENTRY"
        }
      },
      "CONTEXT_ENTRY":{
        "__deferred":{
          "uri":"/idmrestapi/v2/service/
TaskCollection(InstanceID='1x3',SAP__Origin='IDM')/CONTEXT_ENTRY"
        }
      }
    }
  }
}

```

```
}  
}  
]  
}  
}
```

Related Information

[Approval Processing \[page 34\]](#)

[Attestation Access URIs \[page 937\]](#)

8.15.2 Process an Approval

An open approval can be approved, delegated to another user, or declined by the currently logged-in user, if that user is the currently assigned processor for the approval. Approval operations are executed using the decision service operation. The available decision options for a specific approval request can be retrieved by executing the `DecisionOptions` service operation.

8.15.2.1 Get Valid Decision Options for Approval

Valid decision options can be retrieved by executing the `DecisionOptions` service operation. You must specify the request ID (`InstanceID`) as a parameter. The `DecisionKey` property with the respective value must be specified as a parameter for the decision service operation. The [1x](#) part of the instance ID shows that the request is for an approval task.

URI

```
(* [GET] *)  
URI      = './idmrestapi/v2/service/DecisionOptions'
```

Request Parameters

Parameter	= '?' InstanceID
InstanceID	= 'InstanceID=' RequestId
RequestId	= Edm.String;

Request Example

```
[GET] /idmrestapi/v2/service/DecisionOptions?InstanceID='1x4'
```

Response Types

```
Content-Type: application/json
```

HTTPS Response Codes

- 200 - OK
- 401 - Unauthorized
If the user is not authenticated.
- 405 - Method Not Allowed
If the URI was not called with a GET request.

Response Example

```
{
  "d": [
    {
      "SAP__Origin": "IDM",
      "InstanceID": "1x4",
      "DecisionKey": "APPROVE",
      "DecisionText": "Approve",
      "CommentMandatory": false,
      "Nature": "POSITIVE"
    },
    {
      "SAP__Origin": "IDM",
      "InstanceID": "1x4",
      "DecisionKey": "DECLINE",
      "DecisionText": "Decline",
      "CommentMandatory": false,
      "Nature": "NEGATIVE"
    },
    {
      "SAP__Origin": "IDM",
      "InstanceID": "1x4",
      "DecisionKey": "DELEGATE",
      "DecisionText": "Delegate",
      "CommentMandatory": false,
      "Nature": "NEGATIVE"
    }
  ]
}
```


8.15.2.2 Execute Approval Decision

Get Valid Decision Options describes how to retrieve a list of available decision options. After choosing a decision option, you must specify the **DecisionKey** and the request ID (**InstanceID**) of the approval request as a parameter of the decision service operation. The [Ix](#) part of the **InstanceID** indicates that this is an approval request. You can also specify a comment. The **DecisionOptions** service operation indicates whether the **Comments** parameter is mandatory. If the approval decision succeeds, the approval request (task) is returned to the client.

URI

```
(* [POST] *)  
URI      = './idmrestapi/v2/service/Decision'
```

Request Parameters

```
Parameter = '?' InstanceID '&' DecisionKey '&' SAP__Origin [ '&' Comments '&'  
UserToDelegate '&' NewValidFrom '&' NewValidTo]  
InstanceID = 'InstanceID' = RequestId  
DecisionKey = 'DecisionKey' = DecisionKeyValue  
SAP__Origin = 'SAP__Origin' = 'IDM'  
Comments = 'Comments' = CommentsValue  
UserToDelegate = 'UserToDelegate' = UserMSKEYVALUE  
NewValidFrom = 'NewValidFrom' = NewValidFromValue  
NewValidTo = 'NewValidTo' = NewValidToValue  
RequestId = Edm.String;  
DecisionKeyValue = Edm.String;  
CommentsValue = Edm.String;  
UserMSKEYVALUE = Edm.String;  
NewValidFromValue = Edm.String;  
NewValidToValue = Edm.String;
```

ⓘ Note

The parameters in the brackets ([]) are optional.

ⓘ Note

The parameters **NewValidFrom** and **NewValidTo** can be used independently or combined.

Their values are in milliseconds.

→ Recommendation

It is highly recommended to work only with UTC time.

Request Example 1

```
[POST]idmrestapi/v2/service/Decision?
InstanceID='1x4'&DecisionKey='APPROVE'&SAP__Origin='IDM'
```

Response Example 1

```
HTTPS/1.1 200 OK
{
  "d":{
    {
      "__metadata": {
        "uri": "/idmrestapi/v2/service/
TaskCollection(InstanceID='1x4',SAP__Origin='IDM')",
        "type": "IdmRestModel.Task"
      },
      "SAP__Origin": "IDM",
      "InstanceID": "1x4",
      "CREATED_BY_ID": 89,
      "CreatedBy": "Hans Müller",
      "CreatedOn": "\/Date(1360800000000)\/",
      "REQUESTED_ENTRY_ID": 2441,
      "REQUESTED_ENTRY_NAME": "SAP_ALL-Profile (BCE324)",
      "ExpiryDate": "\/Date(1363392000000)\/",
      "VALID_FROM": "\/Date(1363651200000)\/",
      "VALID_TO": "\/Date(1364601600000)\/",
      "REASON": "Need SDIC Super Admin rights",
      "CONTEXT": {
        "ID": 325,
        "DISPLAY_NAME": "ProjectX",
        "ENTRYTYPE_DISPLAY_NAME": "Project Context"
      }, "REFERENCED_ENTRY": {
        "__deferred": {
          "uri": "/idmrestapi/v2/service/
TaskCollection(InstanceID='1x4',SAP__Origin='IDM')/REFERENCED_ENTRY"
        }
      }, "CONTEXT_ENTRY": {
        "__deferred": {
          "uri": "/idmrestapi/v2/service/
TaskCollection(InstanceID='1x4',SAP__Origin='IDM')/CONTEXT_ENTRY"
        }
      }
    }
  }
}
```

Request Example 2

```
[POST]idmrestapi/v2/service/Decision?
InstanceID='1x4'&DecisionKey='DECLINE'&Comments='Not allowed'&SAP__Origin='IDM'
```

Response Example 2

```
HTTPS/1.1 200 OK{
  "d":{
    "__metadata":{
      "id":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')",
      "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')",
      "type":"TASKPROCESSING.Task"
    },
    "SAP__Origin":"IDM",
    "InstanceID":"1x4",
    "TaskDefinitionID":"1",
    "CreatedBy":"TaskCollectionTestUser",
    "CreatedOn":"\\Date(1660608000000)\\",
    "ExpiryDate":"\\Date(253402258332000)\\",
    "CREATED_BY_ID":2234,
    "REQUESTED_ENTRY_ID":436,
    "REQUESTED_ENTRY_NAME":"RoleForAssignmentValidity",
    "VALID_FROM":null,
    "VALID_TO":null,
    "REASON":null,
    "Status":null,
    "COMPLETED":null,
    "CONTEXT":{
      "__metadata":{
        "type":"IdmRestModel.CONTEXT"
      },
      "CONTEXT_ID":null,
      "CONTEXT_DISPLAY_NAME":null,
      "CONTEXT_ENTRYTYPE_DISPLAY_NAME":null
    },
    "CompletionDeadLine":null,
    "TaskTitle":"Approval Task",
    "TaskName":"Approval Task",
    "OPERATION_ID":1,
    "OPERATION_NAME":"Add",
    "REQUESTED_BY_ID":-3,
    "REQUESTED_BY_NAME":null,
    "REQUESTED_VALID_FROM":null,
    "REQUESTED_VALID_TO":null,
    "MODIFY_REASON":null,
    "TASK_GUID":"F310D68D-C76C-4FE0-A7D2-D0431F390176",
    "CustomAttributeData":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/CustomAttributeData"
      }
    },
    "TaskDefinitionData":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/TaskDefinitionData"
      }
    },
    "Assignments":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/Assignments"
      }
    },
    "REFERENCED_ENTRY":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/REFERENCED_ENTRY"
```

```

    }
  },
  "CONTEXT_ENTRY": {
    "__deferred": {
      "uri": "/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM', InstanceID='1x4')/CONTEXT_ENTRY"
    }
  }
}
}
}
}

```

Request Example 3

```

[POST]/idmrestapi/v2/service/Decision?
InstanceID='1x4'&DecisionKey='DELEGATE'&UserToDelegate='Administrator'&SAP__Origin='IDM'

```

Response Example 3

```

HTTPS/1.1 200 OK{
  "d": {
    "__metadata": {
      "id": "/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM', InstanceID='1x4')",
      "uri": "/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM', InstanceID='1x4')",
      "type": "TASKPROCESSING.Task"
    },
    "SAP__Origin": "IDM",
    "InstanceID": "1x4",
    "TaskDefinitionID": "1",
    "CreatedBy": "TaskCollectionTestUser",
    "CreatedOn": "\Date(1660608000000)\",
    "ExpiryDate": "\Date(253402258332000)\",
    "CREATED_BY_ID": 2156,
    "REQUESTED_ENTRY_ID": 401,
    "REQUESTED_ENTRY_NAME": "RoleForApprovalsDelegationTests",
    "VALID_FROM": null,
    "VALID_TO": null,
    "REASON": null,
    "Status": null,
    "COMPLETED": null,
    "CONTEXT": {
      "__metadata": {
        "type": "IdMRestModel.CONTEXT"
      },
      "CONTEXT_ID": null,
      "CONTEXT_DISPLAY_NAME": null,
      "CONTEXT_ENTRYTYPE_DISPLAY_NAME": null
    },
    "CompletionDeadLine": null,
    "TaskTitle": "Approval Task",
    "TaskName": "Approval Task",
    "OPERATION_ID": 1,
    "OPERATION_NAME": "Add",
    "REQUESTED_BY_ID": -3,
    "REQUESTED_BY_NAME": null,
  }
}

```

```

    "REQUESTED_VALID_FROM":null,
    "REQUESTED_VALID_TO":null,
    "MODIFY_REASON":null,
    "TASK_GUID":"F310D68D-C76C-4FE0-A7D2-D0431F390176",
    "CustomAttributeData":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/CustomAttributeData"
      }
    },
    "TaskDefinitionData":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/TaskDefinitionData"
      }
    },
    "Assignments":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/Assignments"
      }
    },
    "REFERENCED_ENTRY":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/REFERENCED_ENTRY"
      }
    },
    "CONTEXT_ENTRY":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/CONTEXT_ENTRY"
      }
    }
  }
}

```

Request Example 4

```

[POST]/idmrestapi/v2/service/Decision?
InstanceID='1x4'&DecisionKey='APPROVE'&NewValidFrom='1660639714079'&NewValidTo='1
660985314079'&SAP__Origin='IDM'

```

→ Tip

When you specify a value for the parameter `NewValidFrom`, the time will be automatically set to 00:00:00.

When you specify a value for `NewValidTo`, the time will be automatically set to 23:59:59.

Response Example 4

```

HTTPS/1.1 200 OK{
  {
    "d":{

```

```

    "__metadata":{
      "id":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')",
      "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')",
      "type":"TASKPROCESSING.Task"
    },
    "SAP__Origin":"IDM",
    "InstanceID":"1x4",
    "TaskDefinitionID":"1",
    "CreatedBy":"TaskCollectionTestUser",
    "CreatedOn":"\\/Date(1660608000000)\\/",
    "ExpiryDate":"\\/Date(253402258332000)\\/",
    "CREATED_BY_ID":2145,
    "REQUESTED_ENTRY_ID":436,
    "REQUESTED_ENTRY_NAME":"RoleForAssignmentValidity",
    "VALID_FROM":"\\/Date(1660658441338)\\/",
    "VALID_TO":"\\/Date(1660917641338)\\/",
    "REASON":null,
    "Status":null,
    "COMPLETED":null,
    "CONTEXT":{
      "__metadata":{
        "type":"IdMRestModel.CONTEXT"
      },
      "CONTEXT_ID":null,
      "CONTEXT_DISPLAY_NAME":null,
      "CONTEXT_ENTRYTYPE_DISPLAY_NAME":null
    },
    "CompletionDeadLine":null,
    "TaskTitle":"Approval Task",
    "TaskName":"Approval Task",
    "OPERATION_ID":1,
    "OPERATION_NAME":"Add",
    "REQUESTED_BY_ID":-3,
    "REQUESTED_BY_NAME":null,
    "REQUESTED_VALID_FROM":"\\/Date(1660639714079)\\/",
    "REQUESTED_VALID_TO":"\\/Date(1660985314079)\\/",
    "MODIFY_REASON":null,
    "TASK_GUID":"F310D68D-C76C-4FE0-A7D2-D0431F390176",
    "CustomAttributeData":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/CustomAttributeData"
      }
    },
    "TaskDefinitionData":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/TaskDefinitionData"
      }
    },
    "Assignments":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/Assignments"
      }
    },
    "REFERENCED_ENTRY":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/REFERENCED_ENTRY"
      }
    },
    "CONTEXT_ENTRY":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/CONTEXT_ENTRY"
      }
    }
  }
}

```

```
}  
  }  
} }
```

Response Types

Content-Type: application/json

HTTPS Response Codes

- 200 - OK
- 401 - Unauthorized
If the user is not authenticated.
- 403 - Forbidden
If the user is not authorized to process these open approvals.
- 404 - Not Found
If the requested approval item does not exist.
- 405 - Method Not Allowed
If the URI was not called with a POST request.

8.16 Attestation Access URIs

You can use the following URIs to access attestation items of SAP Identity Management, for example, user assignments to specific roles that need to be confirmed periodically by an authorized person. The attester can confirm or reject the user assignment. He or she can also delegate this responsibility to another attester. The attestation tasks are part of the same resource as the approval tasks.

Related Information

[Approval Access URIs \[page 926\]](#)

8.16.1 Get Attestation Requests for Current User

The GET request returns a list containing attestation items assigned to the logged-in user. The list includes all of the information about the attested item that is needed for an attester to decide whether to confirm, reject, or delegate the item. The list can also contain information about the approval task.

URI

```
(* [GET] *)  
URI      = './idmrestapi/v2/service/TaskCollection'
```

Request Example

```
[GET]/idmrestapi/v2/service/TaskCollection
```

Request Example for Getting One Attestation Task

The currently logged-in user can see one specific attestation task identified by the **InstanceID** parameter. The task contains all the attestation resources the user needs for the attestation. The [2x](#) part of the instance ID indicates that this is an attestation task.

- Request Example

```
[GET]/idmrestapi/v2/service/  
TaskCollection(InstanceID='2x11',SAP__Origin='IDM')
```

Request Example for Getting Assignments

The currently logged-in user can see detailed information about the assignments for all the resources of one specific attestation task identified by **InstanceID**.

```
[GET]/idmrestapi/v2/service/TaskCollection(InstanceID='2x11',SAP__Origin='IDM')/?  
$expand=Assignments
```


Request Example for Getting Data for All Custom Attributes

The currently logged-in user can see detailed information about the custom attributes for all the resources of one specific attestation task identified by `InstanceID`.

The data contains detailed information about the following attributes:

- `ROLE_NAME`
- `ROLE_DESCRIPTION`
- `LAST_ATTESTED_ON`
- `ASSIGNMENTS_COUNT`
- `STARTED_BY_ID`
- `STARTED_BY`
- `ATTESTATION_TYPE`

```
[GET]/idmrestapi/v2/service/TaskCollection(InstanceID='2x11',SAP__Origin='IDM')/?$expand=CustomAttributeData
```

Response Types

```
Content-Type: application/json
```

HTTPS Response Codes

- **200 - OK**
- **401 - Unauthorized**
If the user is not authenticated.
- **403 - Forbidden**
If the user is not authorized to query for attestation.
- **405 - Method Not Allowed**
If the URI was not called with a GET request.

Note

The collection `UIExecutionLink` is not supported as part of the OData model. If you include it in the request, in the HTTPS response you will get a **Not Found** error.

Response Example

```
{  
  "d": {
```

```

"results": [
  {
    "__metadata": {
      "uri": "/idmrestapi/v2/service/
TaskCollection(InstanceID='2x11',SAP__Origin='IDM')",
      "type": "TASKPROCESSING.Task"
    },
    "SAP__Origin": "IDM",
    "TaskDefinitionID": "2",
    "InstanceID": "2x11",
    "CreatedOn": "/Date(1382572800000)/",
    "CreatedBy": "<SYSTEM>",
    "CompletionDeadLine": "/Date(1382659200000)/",
    "ExpiryDate": "/Date(1382659200000)/",
    "TaskTitle": "Attestation for Role/Privilege Administrator",
    "TaskDefinitionData": {
      "__deferred": {
        "uri": "/idmrestapi/v2/service/
TaskCollection(InstanceID='2x11',SAP__Origin='IDM')/TaskDefinitionData"
      }
    },
    "CustomAttributeData": {
      "results": [
        {
          "__metadata": {
            "uri": "/idmrestapi/v2/service/
CustomAttribute(InstanceID='2x11',Name='ROLE_NAME',SAP__Origin='IDM')",
            "type": "TASKPROCESSING.CustomAttribute"
          },
          "SAP__Origin": "IDM",
          "InstanceID": "2x11",
          "Name": "ROLE_NAME",
          "Value": "Administrator"
        },
        {
          "__metadata": {
            "uri": "/idmrestapi/v2/service/
CustomAttribute(InstanceID='2x11',Name='ROLE_DESCRIPTION',SAP__Origin='IDM')",
            "type": "TASKPROCESSING.CustomAttribute"
          },
          "SAP__Origin": "IDM",
          "InstanceID": "2x11",
          "Name": "ROLE_DESCRIPTION",
          "Value": "Administrators have rights to access additional
resources."
        },
        {
          "__metadata": {
            "uri": "/idmrestapi/v2/service/
CustomAttribute(InstanceID='2x11',Name='LAST_ATTESTED_ON',SAP__Origin='IDM')",
            "type": "TASKPROCESSING.CustomAttribute"
          },
          "SAP__Origin": "IDM",
          "InstanceID": "2x11",
          "Name": "LAST_ATTESTED_ON",
          "Value": "2013-10-24T00:00"
        },
        {
          "__metadata": {
            "uri": "/idmrestapi/v2/service/
CustomAttribute(InstanceID='2x11',Name='ASSIGNMENTS_COUNT',SAP__Origin='IDM')",
            "type": "TASKPROCESSING.CustomAttribute"
          },
          "SAP__Origin": "IDM",
          "InstanceID": "2x11",
          "Name": "ASSIGNMENTS_COUNT",
          "Value": "1"
        }
      ]
    }
  ]
}

```

```

    {
      "__metadata": {
        "uri": "/idmrestapi/v2/service/
CustomAttribute(InstanceID='2x11',Name='STARTED_BY_ID',SAP__Origin='IDM')",
        "type": "TASKPROCESSING.CustomAttribute"
      },
      "SAP__Origin": "IDM",
      "InstanceID": "2x11",
      "Name": "STARTED_BY_ID",
      "Value": "-1"
    },
    {
      "__metadata": {
        "uri": "/idmrestapi/v2/service/
CustomAttribute(InstanceID='2x11',Name='STARTED_BY',SAP__Origin='IDM')",
        "type": "TASKPROCESSING.CustomAttribute"
      },
      "SAP__Origin": "IDM",
      "InstanceID": "2x11",
      "Name": "STARTED_BY",
      "Value": "<SYSTEM>"
    },
    {
      "__metadata": {
        "uri": "/idmrestapi/v2/service/
CustomAttribute(InstanceID='2x11',Name='ATTESTATION_TYPE',SAP__Origin='IDM')",
        "type": "TASKPROCESSING.CustomAttribute"
      },
      "SAP__Origin": "IDM",
      "InstanceID": "2x11",
      "Name": "ATTESTATION_TYPE",
      "Value": "1"
    }
  ]
},
"UIExecutionLink": {
  "__deferred": {
    "uri": "/idmrestapi/v2/service/
TaskCollection(InstanceID='2x11',SAP__Origin='IDM')/UIExecutionLink"
  }
},
"Assignments": {
  "results": [
    {
      "__metadata": {
        "uri": "/idmrestapi/v2/service/
Assignments(InstanceID='2x11',LINK_ID=31,SAP__Origin='IDM')",
        "type": "TASKPROCESSING.Assignments"
      },
      "InstanceID": 2x11,
      "LINK_ID": 31,
      "SAP__Origin": "IDM",
      "LAST_ATTESTED_BY_DISPLAYNAME": "John Miller",
      "LAST_ATTESTED_BY_ID": 12,
      "LAST_ATTESTED_ON": "/Date(1379548800000)/",
      "USER_ID": 16,
      "USER_DISPLAYNAME": "Julie Armstrong",
      "VALID_TO": null,
      "STATE": "0",
      "CONTEXT_ID": 0,
      "CONTEXT_DISPLAY_NAME": null
    }
  ]
}
}

```

Related Information

[Approval Access URIs \[page 926\]](#)

8.16.2 Process an Attestation

An attestation can be confirmed, rejected, or delegated by the currently logged-in user, if that user is the currently assigned processor for the attestation. Attestation operations are executed using the decision service operation. The available decision options for a specific attestation request can be retrieved by executing the `DecisionOptions` service operation.

8.16.2.1 Get Valid Decision Options for Attestation

Valid decision options for attestation can be retrieved by executing the `DecisionOptions` service operation. You must specify the request ID (`InstanceID`) as a parameter. The `DecisionKey` property with the respective value must be specified as a parameter for the decision service operation. The [2x](#) part of the `instanceID` shows that the request is for an attestation task.

URI

```
(* [GET] *)  
URI = './idmrestapi/v2/service/DecisionOptions'
```

Request Parameters

Parameter	= '?' InstanceID
InstanceID	= 'InstanceID=' RequestId
RequestId	= Edm.String;

Request Example

```
[GET] /idmrestapi/v2/service/DecisionOptions?InstanceID='2x11'
```

Response Types

Content-Type: application/json

HTTPS Response Codes

- 200 - OK
- 401 - Unauthorized
If the user is not authenticated.
- 405 - Method Not Allowed
If the URI was not called with a GET request.

Response Example

```
{
  "d": [
    {
      "SAP__Origin": "IDM",
      "InstanceID": "2x11",
      "DecisionKey": "DELEGATE",
      "DecisionText": "Delegate",
      "CommentMandatory": false,
      "Nature": "NEGATIVE"
    },
    {
      "SAP__Origin": "IDM",
      "InstanceID": "2x11",
      "DecisionKey": "ATTEST",
      "DecisionText": "Attest",
      "CommentMandatory": false,
      "Nature": "POSITIVE"
    }
  ]
}
```

8.16.2.2 Execute Attestation Decision

After you have selected the appropriate `DecisionOptions`, `ATTEST` or `DELEGATE`, you can go on to perform the attestation on a specific task, which can be uniquely identified by its instance ID. The value of the `instanceID` parameter starts with `2x`, which indicates that this is an attestation task.

URI

```
(* [POST] *)  
URI = './idmrestapi/v2/service/Decision'
```

Request Parameters

```
Parameters = '?' InstanceID '&' DecisionKey '&' Action '&' LinkId '&  
DelegateId [ '&' Comments ]  
InstanceID = 'InstanceID=' RequestId  
SAP_Origin = 'SAP_Origin=' 'IDM'  
DecisionKey = 'DecisionKey=' DecisionKeyValue  
Action = 'Action=' ActionValue  
LinkId = 'LinkId=' LinkIdValue  
DelegateId = 'DelegateId=' DelegateIdValue  
Comments = 'Comments=' CommentsValue  
RequestId = Edm.String;  
DecisionKeyValue = Edm.String;  
ActionValue = Edm.String;  
LinkIdValue = Edm.String;  
DelegateIdValue = Edm.String;  
CommentsValue = Edm.String;
```

Note

The parameters in the brackets ([]) are optional.

Request Example for Positive Attestation

The attester confirms that the attested role belongs to the attested user with the following settings:

- DecisionKey: ATTEST
- Action: CERTIFY
- LinkId: ID of one particular resource to be attested

```
[POST] idmrestapi/v2/service/Decision?InstanceID='2x11'&  
SAP__Origin='IDM'&DecisionKey='ATTEST'&Action='CERTIFY'&LinkId='31'
```

Request Example for Negative Attestation

The attester rejects that the attested role belongs to the attested user with the following settings:

- DecisionKey: ATTEST
- Action: REJECT
- LinkId: ID of one particular resource to be attested

```
[POST] idmrestapi/v2/service/Decision?InstanceID='2x11'&
SAP__Origin='IDM'&DecisionKey='ATTEST'&Action='REJECT'&LinkId='31'
```

Request Example for Delegating Attestation

The attester is not sure if the attested role belongs to the attested user, so he delegates the attestation of one of the assignments to another attester with the following settings:

- DecisionKey: DELEGATE
- Delegateld: ID of the user to whom the attestation is to be delegated
- LinkId: ID of one particular resource to be attested

```
[POST] idmrestapi/v2/service/Decision?InstanceID='2x11'&
SAP__Origin='IDM'&DecisionKey='DELEGATE'&DelegateId='30'&LinkId='900'
```

Note

The **LinkId** parameter in this example specifies which assignment to be delegated for attestation.

Response Types

Content-Type: application/json

HTTPS Response Codes

- 200 - OK
- 401 - Unauthorized
If the user is not authenticated.
- 403 - Forbidden
If the user is not authorized to process this task.
- 404 - Not Found
If the requested attestation item does not exist.
- 405 - Method Not Allowed

If the URI was not called with a POST request.

Response Example

```
{
  "d": {
    "metadata": {
      "uri": "/idmrestapi/v2/service/
TaskCollection(InstanceID='2x11',SAP__Origin='IDM')",
      "type": "TASKPROCESSING.Task"
    },
    "SAP__Origin": "IDM",
    "TaskDefinitionID": "2",
    "InstanceID": "2x11",
    "TaskTitle": "Attestation"
  }
}
```

8.16.3 Get a History of Attestations

You can use this request to display a history log of previous attestations. The log can give you information about each attester, the resource the attester had to attest or delegate, and his or her action.

URI

```
(* [GET] *)
URI      = 'idmrestapi/v2/service/TaskHistory'
```

Request Parameters

You can customize the history log using the following parameters:

- **Scope**
The Scope parameter specifies what part of the attestation history is displayed for the user. It can have the following three values:
 - [Global](#)
The request returns information about all of the attestations that were previously executed. This is the default value.
 - [User](#)
The request returns information about all of the attestations done for a specific user. If the Scope parameter has the value [User](#), you must also specify the parameter [UserId](#).
 - [Assignment](#)

The request returns information about all of the actions that were performed on a specific attestation resource. If you use the [Assignment](#) value, you must specify the `LinkId` parameter. The `LinkId` parameter identifies a specific attestation resource. For example, it indicates the connection between a user and a role.

Note

If the `Scope` parameter is omitted in the URI, the system assumes that `Scope=Global`.

- **TaskType**

Currently, this parameter has only one possible value: [Attestation](#).

Note

If the `TaskType` parameter is omitted in the URI, the system assumes that `TaskType=Attestation`.

Parameter	= '?' Scope '&' TaskType '&' UserId '&' LinkId
Scope	= 'Scope=' ScopeValue
TaskType	= 'TaskType=' Attestation
UserId	= 'UserId=' UserIdValue
LinkId	= 'LinkId=' LinkIdValue
ScopeValue	= Global User Assignment
UserIdValue	= Edm.Int32;
LinkIdValue	= Edm.Int32;

Request Example

```
[GET] idmrestapi/v2/service/TaskHistory
```

With this request, the response contains all of the information for the history of attestations.

Request Example with Global Scope

```
[GET] idmrestapi/v2/service/TaskHistory?Scope=Global&TaskType=Attestation
```

With this request, the response contains all of the information for the history of attestations.

Request Example with User Scope

```
[GET] idmrestapi/v2/service/TaskHistory?  
Scope=User&UserId=16&TaskType=Attestation
```

With this request, the response contains all of the attestations done for the user with the ID 16. The name of this user is Julie Armstrong.

Request Example with Assignment Scope

```
[GET] idmrestapi/v2/service/TaskHistory?
Scope=Assignment&LinkId=31&TaskType=Attestation
```

With this request, the response contains all the attestations done for the role *Administrator* and the user *Julie Armstrong*. The value *31* for the *LinkId* parameter indicates that the user Julie Armstrong has the *Administrator* role.

Response Types

```
Content-Type: application/json
```

HTTPS Response Codes

- 200 - OK
- 401 - Unauthorized
If the user is not authenticated.
- 403 - Forbidden
If the user is not authorized to query for a history log.
- 405 - Method Not Allowed
If the URI was not called with a GET request.

Response Example

```
{
  d:
  {
    results:
    [2]
    0:
    {
      __metadata:
      {
        uri: "/idmrestapi/v2/service/TaskHistory(LINK_ID='31',OPERATION='Attestation:
        confirmed')"
        type: "TASKPROCESSING.TaskHistoryType"
      }
    }
    -
    LINK_ID: 31
    USER_DISPLAYNAME: "Julie Armstrong"
    USER_ID: 16
    ROLE_DISPLAYNAME: "Administrator"
    ROLE_ID: 26
    CONTEXT_DISPLAYNAME: null
    CONTEXT_ID: 0
  }
}
```

```

ASSIGNED_DATE: "/Date(1377561600000)/"
VALID_FROM: null
VALID_TO: null
RESPONSIBLE_DISPLAYNAME: "Donna Moore"
RESPONSIBLE_ID: 12
OPERATION: "Attestation: confirmed"
REASON: "Julie needs the Administrator role for some settings."
AUDIT_DATE: "/Date(1379548800000)/"
}
-
1:
{
  __metadata:
  {
    uri: "/idmrestapi/v2/service/TaskHistory(LINK_ID='32',OPERATION='Attestation: confirmed')"
    type: "TASKPROCESSING.TaskHistoryType"
  }
-
LINK_ID: 32
USER_DISPLAYNAME: "Michael Adams"
USER_ID: 17
ROLE_DISPLAYNAME: "Manager"
ROLE_ID: 27
CONTEXT_DISPLAYNAME: null
CONTEXT_ID: 0
ASSIGNED_DATE: "/Date(1377561600000)/"
VALID_FROM: null
VALID_TO: null
RESPONSIBLE_DISPLAYNAME: "Donna Moore"
RESPONSIBLE_ID: 12
OPERATION: "Attestation: confirmed"
REASON: "Michael is still a manager in the organization."
AUDIT_DATE: "/Date(1378684800000)/"
}
}
}

```

8.17 Service Operations

The following sections provide an overview of the available service operations of the SAP Identity Management OData service, which can be called on the service root URI.

8.17.1 Get Version Information

This service operation returns version information about the Identity Management REST service. This information contains a major version number and minor version number. The major version number is increased when an incompatible change was done. The minor number is increased for every release where compatible changes were done.

Clients can use this information to check if the major or minor version number is higher or equals to a version number that is required by a certain feature. See also *Compatibility*.

URI

```
(* [GET] *)  
URI = './idmrestapi/v2/service/GetVersion';
```

Request Parameters

None.

Request Example

```
GET /idmrestapi/v2/service/GetVersion  HTTPS/1.1
```

Response Types

```
Content-Type: application/json
```

HTTPS Response Codes

- 200 - OK

Response Example

```
{  
  "d": {  
    "VersionInformation": {  
      "MAJOR_VERSION": 2,  
      "MINOR_VERSION": 0  
    }  
  }  
}
```

Related Information

[Compatibility \[page 839\]](#)

8.17.2 Get Valid Decision Options for Approval

Valid decision options can be retrieved by executing the `DecisionOptions` service operation. You must specify the request ID (`InstanceID`) as a parameter. The `DecisionKey` property with the respective value must be specified as a parameter for the decision service operation. The [1x](#) part of the instance ID shows that the request is for an approval task.

URI

```
(* [GET] *)  
URI      = './idmrestapi/v2/service/DecisionOptions'
```

Request Parameters

Parameter	= '?' InstanceID
InstanceID	= 'InstanceID=' RequestId
RequestId	= Edm.String;

Request Example

```
[GET] /idmrestapi/v2/service/DecisionOptions?InstanceID='1x4'
```

Response Types

```
Content-Type: application/json
```

HTTPS Response Codes

- 200 - OK
- 401 - Unauthorized
If the user is not authenticated.
- 405 - Method Not Allowed
If the URI was not called with a GET request.

Response Example

```
{
  "d": [
    {
      "SAP__Origin": "IDM",
      "InstanceID": "1x4",
      "DecisionKey": "APPROVE",
      "DecisionText": "Approve",
      "CommentMandatory": false,
      "Nature": "POSITIVE"
    },
    {
      "SAP__Origin": "IDM",
      "InstanceID": "1x4",
      "DecisionKey": "DECLINE",
      "DecisionText": "Decline",
      "CommentMandatory": false,
      "Nature": "NEGATIVE"
    },
    {
      "SAP__Origin": "IDM",
      "InstanceID": "1x4",
      "DecisionKey": "DELEGATE",
      "DecisionText": "Delegate",
      "CommentMandatory": false,
      "Nature": "NEGATIVE"
    }
  ]
}
```

8.17.3 Execute Approval Decision

Get Valid Decision Options describes how to retrieve a list of available decision options. After choosing a decision option, you must specify the **DecisionKey** and the request ID (**InstanceID**) of the approval request as a parameter of the decision service operation. The *1x* part of the **InstanceID** indicates that this is an approval request. You can also specify a comment. The **DecisionOptions** service operation indicates whether the **Comments** parameter is mandatory. If the approval decision succeeds, the approval request (task) is returned to the client.

URI

```
(* [POST] *)  
URI = './idmrestapi/v2/service/Decision'
```

Request Parameters

```
Parameter = '?' InstanceID '&' DecisionKey '&' SAP__Origin [ '&' Comments '&'  
UserToDelegate '&' NewValidFrom '&' NewValidTo]  
InstanceID = 'InstanceID' = RequestId  
DecisionKey = 'DecisionKey' = DecisionKeyValue  
SAP__Origin = 'SAP__Origin' = 'IDM'  
Comments = 'Comments' = CommentsValue  
UserToDelegate = 'UserToDelegate' = UserMSKEYVALUE  
NewValidFrom = 'NewValidFrom' = NewValidFromValue  
NewValidTo = 'NewValidTo' = NewValidToValue  
RequestId = Edm.String;  
DecisionKeyValue = Edm.String;  
CommentsValue = Edm.String;  
UserMSKEYVALUE = Edm.String;  
NewValidFromValue = Edm.String;  
NewValidToValue = Edm.String;
```

ⓘ Note

The parameters in the brackets ([]) are optional.

ⓘ Note

The parameters `NewValidFrom` and `NewValidTo` can be used independently or combined.

Their values are in milliseconds.

→ Recommendation

It is highly recommended to work only with UTC time.

Request Example 1

```
[POST]idmrestapi/v2/service/Decision?  
InstanceID='1x4'&DecisionKey='APPROVE'&SAP__Origin='IDM'
```

Response Example 1

```

HTTPS/1.1 200 OK
{
  "d":{
    {
      "__metadata": {
        "uri": "/idmrestapi/v2/service/
TaskCollection(InstanceID='1x4',SAP__Origin='IDM')",
        "type": "IdMRestModel.Task"
      },
      "SAP__Origin": "IDM",
      "InstanceID": "1x4",
      "CREATED_BY_ID" : 89,
      "CreatedBy": "Hans Müller",
      "CreatedOn": "\\Date(1360800000000)\\",
      "REQUESTED_ENTRY_ID": 2441,
      "REQUESTED_ENTRY_NAME": "SAP_ALL-Profile (BCE324)",
      "ExpiryDate": "\\Date(1363392000000)\\",
      "VALID_FROM": "\\Date(1363651200000)\\",
      "VALID_TO": "\\Date(1364601600000)\\",
      "REASON": "Need SDIC Super Admin rights",
      "CONTEXT": {
        "ID": 325,
        "DISPLAY_NAME": "ProjectX",
        "ENTRYTYPE_DISPLAY_NAME": "Project Context"
      }, "REFERENCED_ENTRY": {
        "__deferred": {
          "uri": "/idmrestapi/v2/service/
TaskCollection(InstanceID='1x4',SAP__Origin='IDM')/REFERENCED_ENTRY"
        }
      }, "CONTEXT_ENTRY": {
        "__deferred": {
          "uri": "/idmrestapi/v2/service/
TaskCollection(InstanceID='1x4',SAP__Origin='IDM')/CONTEXT_ENTRY"
        }
      }
    }
  }
}

```

Request Example 2

```

[POST]idmrestapi/v2/service/Decision?
InstanceID='1x4'&DecisionKey='DECLINE'&Comments='Not allowed'&SAP__Origin='IDM'

```

Response Example 2

```

HTTPS/1.1 200 OK{
  "d":{
    "__metadata":{
      "id":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')",
      "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')",
      "type":"TASKPROCESSING.Task"
    },
    "SAP__Origin":"IDM",

```



```

    "InstanceID": "1x4",
    "TaskDefinitionID": "1",
    "CreatedBy": "TaskCollectionTestUser",
    "CreatedOn": "\Date(1660608000000)\",
    "ExpiryDate": "\Date(253402258332000)\",
    "CREATED_BY_ID": 2234,
    "REQUESTED_ENTRY_ID": 436,
    "REQUESTED_ENTRY_NAME": "RoleForAssignmentValidity",
    "VALID_FROM": null,
    "VALID_TO": null,
    "REASON": null,
    "Status": null,
    "COMPLETED": null,
    "CONTEXT": {
      "__metadata": {
        "type": "IdMRestModel.CONTEXT"
      },
      "CONTEXT_ID": null,
      "CONTEXT_DISPLAY_NAME": null,
      "CONTEXT_ENTRYTYPE_DISPLAY_NAME": null
    },
    "CompletionDeadLine": null,
    "TaskTitle": "Approval Task",
    "TaskName": "Approval Task",
    "OPERATION_ID": 1,
    "OPERATION_NAME": "Add",
    "REQUESTED_BY_ID": -3,
    "REQUESTED_BY_NAME": null,
    "REQUESTED_VALID_FROM": null,
    "REQUESTED_VALID_TO": null,
    "MODIFY_REASON": null,
    "TASK_GUID": "F310D68D-C76C-4FE0-A7D2-D0431F390176",
    "CustomAttributeData": {
      "__deferred": {
        "uri": "/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/CustomAttributeData"
      }
    },
    "TaskDefinitionData": {
      "__deferred": {
        "uri": "/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/TaskDefinitionData"
      }
    },
    "Assignments": {
      "__deferred": {
        "uri": "/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/Assignments"
      }
    },
    "REFERENCED_ENTRY": {
      "__deferred": {
        "uri": "/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/REFERENCED_ENTRY"
      }
    },
    "CONTEXT_ENTRY": {
      "__deferred": {
        "uri": "/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/CONTEXT_ENTRY"
      }
    }
  }
}

```

Request Example 3

```
[POST]/idmrestapi/v2/service/Decision?
InstanceID='1x4'&DecisionKey='DELEGATE'&UserToDelegate='Administrator'&SAP__Origin='IDM'
```

Response Example 3

```
HTTPS/1.1 200 OK{
  "d":{
    "__metadata":{
      "id":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')",
      "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')",
      "type":"TASKPROCESSING.Task"
    },
    "SAP__Origin":"IDM",
    "InstanceID":"1x4",
    "TaskDefinitionID":"1",
    "CreatedBy":"TaskCollectionTestUser",
    "CreatedOn":"\\Date(1660608000000)\\",
    "ExpiryDate":"\\Date(253402258332000)\\",
    "CREATED_BY_ID":2156,
    "REQUESTED_ENTRY_ID":401,
    "REQUESTED_ENTRY_NAME":"RoleForApprovalsDelegationTests",
    "VALID_FROM":null,
    "VALID_TO":null,
    "REASON":null,
    "Status":null,
    "COMPLETED":null,
    "CONTEXT":{
      "__metadata":{
        "type":"IdmRestModel.CONTEXT"
      },
      "CONTEXT_ID":null,
      "CONTEXT_DISPLAY_NAME":null,
      "CONTEXT_ENTRYTYPE_DISPLAY_NAME":null
    },
    "CompletionDeadLine":null,
    "TaskTitle":"Approval Task",
    "TaskName":"Approval Task",
    "OPERATION_ID":1,
    "OPERATION_NAME":"Add",
    "REQUESTED_BY_ID":-3,
    "REQUESTED_BY_NAME":null,
    "REQUESTED_VALID_FROM":null,
    "REQUESTED_VALID_TO":null,
    "MODIFY_REASON":null,
    "TASK_GUID":"F310D68D-C76C-4FE0-A7D2-D0431F390176",
    "CustomAttributeData":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/CustomAttributeData"
      }
    },
    "TaskDefinitionData":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/TaskDefinitionData"
```

```

    }
  },
  "Assignments": {
    "__deferred": {
      "uri": "/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/Assignments"
    }
  },
  "REFERENCED_ENTRY": {
    "__deferred": {
      "uri": "/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/REFERENCED_ENTRY"
    }
  },
  "CONTEXT_ENTRY": {
    "__deferred": {
      "uri": "/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/CONTEXT_ENTRY"
    }
  }
}
}
}

```

Request Example 4

```

[POST]/idmrestapi/v2/service/Decision?
InstanceID='1x4'&DecisionKey='APPROVE'&NewValidFrom='1660639714079'&NewValidTo='1
660985314079'&SAP__Origin='IDM'

```

→ Tip

When you specify a value for the parameter `NewValidFrom`, the time will be automatically set to 00:00:00.

When you specify a value for `NewValidTo`, the time will be automatically set to 23:59:59.

Response Example 4

```

HTTPS/1.1 200 OK{
  {
    "d": {
      "__metadata": {
        "id": "/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')",
        "uri": "/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')",
        "type": "TASKPROCESSING.Task"
      },
      "SAP__Origin": "IDM",
      "InstanceID": "1x4",
      "TaskDefinitionID": "1",
      "CreatedBy": "TaskCollectionTestUser",
      "CreatedOn": "\\Date(1660608000000)\\",
      "ExpiryDate": "\\Date(253402258332000)\\",
      "CREATED_BY_ID": 2145,
    }
  }
}

```

```

    "REQUESTED_ENTRY_ID":436,
    "REQUESTED_ENTRY_NAME":"RoleForAssignmentValidity",
    "VALID_FROM":"\\/Date(1660658441338)\\/",
    "VALID_TO":"\\/Date(1660917641338)\\/",
    "REASON":null,
    "Status":null,
    "COMPLETED":null,
    "CONTEXT":{
      "__metadata":{
        "type":"IdMRestModel.CONTEXT"
      },
      "CONTEXT_ID":null,
      "CONTEXT_DISPLAY_NAME":null,
      "CONTEXT_ENTRYTYPE_DISPLAY_NAME":null
    },
    "CompletionDeadLine":null,
    "TaskTitle":"Approval Task",
    "TaskName":"Approval Task",
    "OPERATION_ID":1,
    "OPERATION_NAME":"Add",
    "REQUESTED_BY_ID":-3,
    "REQUESTED_BY_NAME":null,
    "REQUESTED_VALID_FROM":"\\/Date(1660639714079)\\/",
    "REQUESTED_VALID_TO":"\\/Date(1660985314079)\\/",
    "MODIFY_REASON":null,
    "TASK_GUID":"F310D68D-C76C-4FE0-A7D2-D0431F390176",
    "CustomAttributeData":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/CustomAttributeData"
      }
    },
    "TaskDefinitionData":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/TaskDefinitionData"
      }
    },
    "Assignments":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/Assignments"
      }
    },
    "REFERENCED_ENTRY":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/REFERENCED_ENTRY"
      }
    },
    "CONTEXT_ENTRY":{
      "__deferred":{
        "uri":"/idmrestapi/v2/service/
TaskCollection(SAP__Origin='IDM',InstanceID='1x4')/CONTEXT_ENTRY"
      }
    }
  }
}

```

Response Types

Content-Type: application/json

HTTPS Response Codes

- 200 - OK
- 401 - Unauthorized
If the user is not authenticated.
- 403 - Forbidden
If the user is not authorized to process these open approvals.
- 404 - Not Found
If the requested approval item does not exist.
- 405 - Method Not Allowed
If the URI was not called with a POST request.

8.17.4 Get Valid Decision Options for Attestation

Valid decision options for attestation can be retrieved by executing the `DecisionOptions` service operation. You must specify the request ID (`InstanceID`) as a parameter. The `DecisionKey` property with the respective value must be specified as a parameter for the decision service operation. The [2x](#) part of the `instanceID` shows that the request is for an attestation task.

URI

```
(* [GET] *)  
URI = './idmrestapi/v2/service/DecisionOptions'
```

Request Parameters

Parameter	= '?' InstanceID
InstanceID	= 'InstanceID=' RequestId
RequestId	= Edm.String;

Request Example

```
[GET] /idmrestapi/v2/service/DecisionOptions?InstanceID='2x11'
```

Response Types

Content-Type: application/json

HTTPS Response Codes

- 200 - OK
- 401 - Unauthorized
If the user is not authenticated.
- 405 - Method Not Allowed
If the URI was not called with a GET request.

Response Example

```
{
  "d": [
    {
      "SAP__Origin": "IDM",
      "InstanceID": "2x11",
      "DecisionKey": "DELEGATE",
      "DecisionText": "Delegate",
      "CommentMandatory": false,
      "Nature": "NEGATIVE"
    },
    {
      "SAP__Origin": "IDM",
      "InstanceID": "2x11",
      "DecisionKey": "ATTEST",
      "DecisionText": "Attest",
      "CommentMandatory": false,
      "Nature": "POSITIVE"
    }
  ]
}
```

8.17.5 Execute Attestation Decision

After you have selected the appropriate `DecisionOptions`, `ATTEST` or `DELEGATE`, you can go on to perform the attestation on a specific task, which can be uniquely identified by its instance ID. The value of the `instanceID` parameter starts with `2x`, which indicates that this is an attestation task.

URI

```
(* [POST] *)
URI      = './idmrestapi/v2/service/Decision'
```

Request Parameters

```
Parameters      = '?' InstanceID '&' DecisionKey '&' Action '&' LinkId '&'
DelegateId [ '&' Comments ]
InstanceID      = 'InstanceID=' RequestId
SAP_Origin      = 'SAP_Origin=' 'IDM'
DecisionKey     = 'DecisionKey=' DecisionKeyValue
Action          = 'Action=' ActionValue
LinkId          = 'LinkId=' LinkIdValue
DelegateId      = 'DelegateId=' DelegateIdValue
Comments        = 'Comments=' CommentsValue
RequestId       = Edm.String;
DecisionKeyValue = Edm.String;
ActionValue     = Edm.String;
LinkIdValue     = Edm.String;
DelegateIdValue = Edm.String;
CommentsValue   = Edm.String;
```

Note

The parameters in the brackets ([]) are optional.

Request Example for Positive Attestation

The attester confirms that the attested role belongs to the attested user with the following settings:

- DecisionKey: ATTEST
- Action: CERTIFY
- LinkId: ID of one particular resource to be attested

```
[POST] idmrestapi/v2/service/Decision?InstanceID='2x11'&
SAP__Origin='IDM'&DecisionKey='ATTEST'&Action='CERTIFY'&LinkId='31'
```

Request Example for Negative Attestation

The attester rejects that the attested role belongs to the attested user with the following settings:

- DecisionKey: ATTEST
- Action: REJECT
- LinkId: ID of one particular resource to be attested

```
[POST] idmrestapi/v2/service/Decision?InstanceID='2x11'&
SAP__Origin='IDM'&DecisionKey='ATTEST'&Action='REJECT'&LinkId='31'
```

Request Example for Delegating Attestation

The attester is not sure if the attested role belongs to the attested user, so he delegates the attestation of one of the assignments to another attester with the following settings:

- DecisionKey: DELEGATE
- Delegateld: ID of the user to whom the attestation is to be delegated
- LinkId: ID of one particular resource to be attested

```
[POST] idmrestapi/v2/service/Decision?InstanceID='2x11'&
SAP__Origin='IDM'&DecisionKey='DELEGATE'&DelegateId='30'&LinkId='900'
```

Note

The **LinkId** parameter in this example specifies which assignment to be delegated for attestation.

Response Types

Content-Type: application/json

HTTPS Response Codes

- 200 - OK
- 401 - Unauthorized
If the user is not authenticated.
- 403 - Forbidden
If the user is not authorized to process this task.
- 404 - Not Found
If the requested attestation item does not exist.
- 405 - Method Not Allowed

If the URI was not called with a POST request.

Response Example

```
{
  "d": {
    "_metadata": {
      "uri": "/idmrestapi/v2/service/
TaskCollection(InstanceID='2x11',SAP__Origin='IDM')",
      "type": "TASKPROCESSING.Task"
    },
    "SAP__Origin": "IDM",
    "TaskDefinitionID": "2",
    "InstanceID": "2x11",
    "TaskTitle": "Attestation"
  }
}
```

8.17.6 GRC Notification

The GRC notification SyncGRCRoles endpoint is used in the enhanced integration between SAP Identity Management 8.0 and SAP Access Control (for GRC solutions) 10.0 or higher that allows you to synchronize business roles between both systems before user provisioning.

When GRC maintained business roles are created or modified, the GRC system calls REST API v2 SyncGRCRoles endpoint to notify SAP Identity Management about the change. This endpoint accepts one mandatory parameter `RepositoryName`.

Request

URI: `http://<host>:<port>/idmrestapi/v2/service/SyncGRCRoles?RepositoryName='GRC'`

HTTP Method: [POST](#)

Permissions: [Role-based permissions required to call the method]

Request Parameters

Parameter	Required	Data Type	Description	Parameter Type
Repository Name	Yes	String	The name of the GRC repository that will trigger its Reset Delta job.	Query string

Request Example

```
POST /idmrestapi/v2/service/SyncGRCRoles?RepositoryName='GRC'
```

Response

Response Status and Error Codes

Code	Reason	Description
200	OK	The AC 10.0 - Initial Load - Business Roles job is started.
400	Bad Request	The AC 10.0 - Initial Load - Business Roles job failed to start or could not be found.
405	Method Not Allowed	The specified method is not allowed, for example, GET method is used instead of POST.

Response Example

```
<?xml version='1.0' encoding='utf-8' ?>
  <d:JobInformation m:type="IdMRestModel.JobInformation">
    <d:JOB_ID m:type="Edm.Int32">303</d:JOB_ID>
  </d:JobInformation>
```

9 Reference

9.1 Reference

9.1.1 Identity Center Properties

Overview of the tabs of the Identity Center properties.

The Identity Center properties contain the following tabs:

General	Configure general parameters of the Identity Center.
Housekeeping	Configure scheduled procedures for housekeeping.
Users	Maintain users for the Identity Center.

Related Information

[General \(Identity Center\) \[page 965\]](#)

[Housekeeping \(Identity Center\) \[page 968\]](#)

[Users \(Identity Center\) \[page 971\]](#)

9.1.1.1 General (Identity Center)

Gives an overview of the [General](#) tab of the Identity Center properties.

The [General](#) tab of the Identity Center properties contains the following fields:

Field	Description
Name	The name of the Identity Center as it was defined when added to the Identity Management Developer Studio.
System Namespace	The namespace given for this Identity Center when the database was installed or upgraded.

Field	Description
<i>Free Text Search Enabled</i>	 Note Only relevant when using Microsoft SQL Server as the database server. This check box is selected if full-text search is enabled for the Identity Center database. For more information about how to enable free-text search, see the document Enabling Full-Text Search.
<i>System Log Level</i>	The log level for the system log. Select the lowest log level to be included in the system log. Only log messages with a log level equal to or higher than the selected log level are included in the system log. You can choose between: • <i>Info</i> • <i>Warning</i> • <i>Error</i> • <i>Fatal</i>  Note Choosing log level higher than <i>Error</i> may degrade the performance of the Identity Center. If you experience performance problems, try lowering the level to <i>Error</i>.
<i>Enable Trace</i>	Select this check box to specify that you want to show the detailed task execution history on the <i>Monitoring</i> tab in the Identity Management User Interface.
<i>Idle Timeout</i>	<i>Job timeouts</i> The idle timeout is the period between the time when a job is scheduled to run and the time when a runtime engine starts processing the job. If this period exceeds the specified number of seconds, the job's state is changed to <i>Timeout</i>.
<i>Start Timeout</i>	<i>Job timeouts/Bootstrap Job timeouts</i> The start timeout is the period between the time when the runtime engine reserves the job and the first time the runtime engine sends status information to the Identity Center database. If this period exceeds the specified number of seconds, the job is rescheduled and the number of start attempts is incremented by one. If the number of start attempts reaches the value specified in the <i>Max. number of start attempts</i> field, the job's state is changed to <i>Error</i>. The value under <i>Bootstrap Job timeouts</i> is used for jobs that have been defined as a <i>Bootstrap job</i> in the job properties.
<i>Execution Timeout</i>	<i>Job timeouts/Bootstrap Job timeouts</i>

Field	Description
	The execution timeout is the period between each time a runtime engine sends status information to the Identity Center during the processing of the job. If this period exceeds the specified number of seconds, the job is rescheduled and the number of start attempts is incremented by one. If the number of start attempts reaches the value specified in the <i>Max. number of start attempts</i> field, the job's state is changed to <i>Error</i>. The value under <i>Bootstrap Job timeouts</i> is used for jobs that have been defined as a <i>Bootstrap job</i> in the job properties.
<i>Max Restarts</i>	Enter the maximum number of times a job can be re-scheduled because of start timeouts and execution timeouts. If this number is exceeded, the job's state is changed to <i>Error</i> , and the job must be restarted manually.
<i>Max Concurrent RT Engines</i>	This is the total number of concurrent runtime engines on the system. This can be used to control the load on the database. The default is 0 (unlimited).
<i>Default Dispatcher</i>	If you want to assign a dispatcher to the imported jobs, you can select one of the defined dispatchers in the list.
<i>Privilege modify policy</i>	This property specifies how to handle modify events on privileges. For more information, see Modify Event Handling
<i>Privilege modify interval</i>	This property holds an interval in seconds, which is used to determine whether an attribute modification triggers provisioning. For more information, see Modify Event Interval
<i>Hide Additional Information in the Assignment Details Dialog Box</i>	Select this checkbox to hide the detailed information in the Assignment Detail dialog box in the Identity Management User Interface. The Assignment Detail dialog box appears when you open a status link of an assignment. By default, this checkbox is deselected.
<i>Max Number of Favorites</i>	This is the maximum number of favorite UI tasks that you can set on the Manage tab in the Identity Management User Interface. The default value is 10.

Related Information

[State \(Job/Action Task\) \[page 1025\]](#)

9.1.1.2 Housekeeping (Identity Center)

Gives an overview of the [Housekeeping](#) tab of the Identity Center properties.

The table contains the following columns:

Column	Description
Enabled	You can enable and disable the procedures by selecting or deselecting the check box in this column.
Procedure	Displays the name of the procedure.
Parameter 1	Displays the name of the first parameter to the procedure. The following parameters are available: • Days to keep Specifies the number of days for which you want to keep an entry. This parameter is available for procedures that clean up logs. • Retention period Specifies the retention period (in days) for which you want to keep attributes of deleted entries. This parameter is only available for Clean up attributes' historic values procedure.
Value 1	The value of the first parameter to the procedure. You can change the value in the column. If you set the value of the Retention period to 0, this means that there is no retention period for the attributes of deleted entries. Thus, if an entry is deleted, all attributes of this entry are deleted when the Clean up attributes' historic values procedure is run.
Parameter 2	The description of the second parameter to the procedure.
Value 2	The value of the second parameter to the procedure. You can change the value in the column.
Schedule rule	The schedule rule defined for the procedure. You can select another schedule rule for the procedure in the column. You can also select one of the scheduled procedures and choose "Run now" from the context menu.

Related Information

[Policy \(Dispatcher\) \[page 977\]](#)

[Dispatchers \[page 67\]](#)

9.1.1.2.1 Scheduled Procedures for Housekeeping

Gives an overview of the scheduled procedures used for housekeeping in the Identity Center.

Column	Description
<i>Dispatcher log: Clean up</i>	Removes log entries from the dispatcher log older than the specified number of days to keep. Log entries containing errors are not removed.
<i>Dispatcher log: Clean up including errors</i>	Removes log entries from the dispatcher log older than the specified number of days to keep. Log entries containing errors are also removed.
<i>System log: Clean up</i>	Removes log entries from the system log older than the specified number of days to keep. Log entries containing warnings errors are not removed.
<i>System log: Clean up including errors</i>	Removes log entries from the system log older than the specified number of days to keep. Log entries containing errors are also removed.
<i>Job log: Clean up</i>	Removes log entries from the job log older than the specified number of days to keep. Log entries containing warnings and errors are not removed.
<i>Job log: Clean up including errors</i>	Removes log entries from the job log older than the specified number of days to keep. Log entries containing errors are also removed.
<i>Check for pending values</i>	Checks if there are any pending future value objects for single value attributes, where the valid from time has past.
<i>Check for expired attributes</i>	Checks for attributes where the specified time for Valid to is reached.
<i>Reconcile dirty entries</i>	Reconciles entries with the "dirty" flag set. See Reconciliation [page 99]
<i>Initiate attestation</i>	Initiates attestation.
<i>Execution log: Clean up</i>	Removes log entries from the execution log older than the specified number of days to keep. Log entries containing warnings and errors are not removed.

Column	Description
<i>Execution log: Clean up including errors</i>	Removes log entries from the execution log older than the specified number of days to keep. Log entries containing errors are also removed.
<i>Repository check</i>	Automatically enables or disables repositories according to the date range configured via the <code>uRepositoryConfig</code> function. See uRepositoryConfig [page 1206]
<i>Clean up attributes' historic values</i>	- Removes attribute entries from <code>mxi_old_values</code> table that are older than the specified number of days to keep (configured on each attribute). - Removes attribute entries of deleted entries from <code>mxi_old_values</code> table that are older than the specified number of days (configured for the Retention period parameter). See Cleaning Up Historic Values in the Identity Store
<i>Clean up old report entries</i>	Removes report entries (generated by Standard Reports [page 780] and HTML Reports) from the <code>mxi_entry</code>, <code>mxi_values</code> and <code>mxi_link</code> tables that are older than the specified number of days to keep (by default, 30 days). You can configure the number of <i>Days to keep</i>. This housekeeping procedure does not clean up audit information. It is not enabled by default.

Related Information

[Housekeeping \(Identity Center\) \[page 968\]](#)

[Configuring Scheduled Procedures for Housekeeping \[page 116\]](#)

[Pending Value Objects \[page 95\]](#)

9.1.1.3 Users (Identity Center)

Gives an overview of the [Users](#) tab of the Identity Center properties.

The table contains the following columns:

Column	Description
User Name	The user's name. A symbol in front of the user name indicates that the user is inactive (not allowed to log on).
Type	This can be one of the following user types. - Developer A regular user that can be given authorization to packages. - Developer Administrator An administrator who is allowed to create package, add users, add identity stores, modify identity store schema (entry types/attributes).
Created by	The name of the user who added this user.
Created Date	The date and time the user was added.
Last Login	The date and time the user last logged on the Identity Center database.

Related Information

[Configuring Identity Management Developer Studio Users \[page 117\]](#)

[Adding an Identity Management Developer Studio User \[page 118\]](#)

[Viewing Details About an Identity Management Developer Studio User \[page 120\]](#)

[Modifying Identity Management Developer Studio User \[page 121\]](#)

[Deactivating an Identity Management Developer Studio User \[page 122\]](#)

[Activating an Identity Management Developer Studio User \[page 123\]](#)

9.1.2 Identity Store Properties

Overview of the tabs of the identity store properties.

The identity store properties contain the following tabs:

General	Configure general parameters of the identity store.

Related Information

[General \(Identity Store\) \[page 972\]](#)

[Password Policy \(Identity Store\) \[page 973\]](#)

9.1.2.1 General (Identity Store)

Gives an overview of the properties on the *General* tab of the identity store.

The *General* tab of the identity store properties contains the following fields:

Field	Description
<i>ID/Name</i>	The field displays the ID and name of the identity store. You modify the name in the console tree.
<i>Automatically create attributes</i>	Select this check box to specify that new attributes can be automatically added to the identity store when a job is executed. If an entry contains an attribute that does not already exist in the identity store, the attribute will be added using default values for identity store attributes. In addition, the attribute is also added as a legal (but not mandatory) attribute for the given entry type. Note This option should normally not be enabled in a production system. Note The new attributes are added automatically with their default properties, that is, they are single-value attributes. If you leave the check box empty, inserting a non-existent attribute, or adding an attribute to an entry type which is not allowed, will fail. In this case, all additional attributes must be added to the identity store manually.
<i>Description</i>	Enter a text describing the identity store.
<i>Max search result size</i>	Enter the maximum number of entries to be returned by a search in the Identity Management User Interface.
<i>User interface repository</i>	Select the repository that is used to indicate attribute ownership when writing attributes to the identity store from the Identity Management User interface.

Field	Description
Show favorite task bar in display tasks	This check box controls behavior in the Identity Management User Interface regarding display tasks for reference attributes like roles or privileges. If you select this check box, the favorite tasks will be available in addition to the "Choose task..." in the display task of the reference attributes. Note To be able to access tasks from the display task of a reference attribute, the user has to have the privilege <code>MX_PRIV:WD:TAB_MANAGE</code>. Note Evaluating the access control of the favorite tasks may have impact on performance. This should be considered before enabling this feature.
Request complete process	Select the process that will be executed when the last assignment request with a given request ID has completed. You can select a process of type Request Process. Note An identity store can only reference one Request complete process, so this task must be able to process all assignment requests. Make sure to follow the defined syntax for the request ID.

Related Information

[Identity Store \[page 74\]](#)

[Configuring General Parameters for an Identity Store \[page 126\]](#)

[Adding a User to the Identity Store \[page 127\]](#)

9.1.2.2 Password Policy (Identity Store)

Field	Description
Enable Password Provisioning	Password provisioning is the process of updating a user's password in several systems based on a master password in the identity store. This requires that the password is stored in the <code>MX_ENCRYPTED_PASSWORD</code>, which is encrypted using a two-way encryption.

Field	Description
	Select this check box to specify that a two-way encrypted version of the password (MX_ENCRYPTED_PASSWORD) should be stored separately to be used for password provisioning, whenever storing the MX_PASSWORD attribute. When writing the MX_ENCRYPTED_PASSWORD from the runtime, you must use the uEncrypt function to ensure that it is encrypted. If you use the Basic authentication method, this checkbox must be selected.
Password Reset Form	Select the form that will be called to perform password reset. The access control of this form defines who has access to this task. This should normally be all anonymous users.
Use Password Dictionary	Select this checkbox to specify that you want to use a password dictionary with words that are not allowed in a password. Maintain the word list manually, or import a word list to use as the password dictionary.
Check Password History	To prevent a user from entering a recently used password, you can specify that a user's new password must be checked against the previous passwords. How many versions of passwords are kept, is decided by the number of revisions that are specified for the MX_PASSWORD, default 3.

Related Information

[Identity Store \[page 74\]](#)

[Configuring the Password Policy of an Identity Store \[page 128\]](#)

[Defining a Password Dictionary \[page 129\]](#)

9.1.3 Dispatcher Properties

Overview of the dispatcher properties.

The dispatcher properties contain the following tabs:

General	Configure/view general properties for the dispatcher - name, runtime engine information, log level, status etc.
Policy	Specify information about a dispatcher's task execution policy.
Jobs	Specify the jobs to be run by the dispatcher.

[Advanced](#)

Specify the custom parameters for the dispatcher. Only visible if you choose the [Advanced](#) button from the [General](#) tab.

Related Information

[General \(Dispatcher\) \[page 975\]](#)

[Advanced \(Dispatcher\) \[page 979\]](#)

[Policy \(Dispatcher\) \[page 977\]](#)

[Jobs \(Dispatcher\) \[page 978\]](#)

[Dispatchers \[page 67\]](#)

9.1.3.1 General (Dispatcher)

The [General](#) tab of the dispatcher properties is used to define general information about the dispatcher, e.g. dispatcher name and log level.

The [General](#) tab of the dispatcher properties contains the following fields:

Field	Description
Name	The field displays the name of the dispatcher. You can modify the name. The name is converted to lowercase, and any spaces (and other special characters) are converted to underscores.
Version	The field displays the dispatcher's version number. It is automatically updated by the dispatcher when connecting to the Identity Center.
Max Concurrent RT Engines	Enter the maximum number of concurrent runtime engines for this dispatcher. The default is <i>0</i>, which means no upper limit. Example: Max concurrent rt engines is set to 5. If there are 10 jobs ready to be executed and 4 already running, the dispatcher will start one job not to exceed the value defined in Max Concurrent RT Engines (5).
Max Loops for RT Engines	Each time a runtime engine has completed processing a job, it will check to see whether there are more jobs waiting to be processed. Each iteration is defined as a loop. This optimizes the job/task execution insofar that the runtime engine does not have to be started for each job/task.

Field	Description
	Enter the maximum number of loops for a runtime engine. This is to avoid problems caused for instance by memory leaks in the custom code of the job. The default is <i>1</i> .
<i>Last Check</i>	The field displays the time when the dispatcher was last checked for scheduled jobs.
<i>Log Level</i>	Select the log level for the dispatcher when logging to the system log. You can choose between: • <i>Info</i> • <i>Warning</i> • <i>Error</i> • <i>Fatal</i> • <i>Debug</i> • <i>Trace</i> The default value is <i>Error</i>.
<i>Status</i>	This field shows the last status reported by the dispatcher, e.g. <i>Stopped</i>, <i>Suspending...</i>, <i>Suspended</i>, <i>Resuming...</i> or <i>Running</i>. Use the <i>Refresh</i> button to update the status. Use the <i>Suspend</i> button to suspend a running dispatcher (a sleep-mode function that sets the dispatcher in the <i>Suspended</i> status). When suspended, the dispatcher service is still running, but its processing is suspended. Use the <i>Resume</i> button to resume a suspended dispatcher (sets the suspended dispatcher in the <i>Running</i> status). The dispatcher resumes the processing. Use the <i>Stop Dispatcher</i> button to stop a dispatcher. Note To start a stopped dispatcher, use the Identity Management Dispatcher Utility.
<i>Advanced</i>	Choose this button to open the <i>Advanced</i> tab of the dispatcher properties, and configure the custom parameters for the dispatcher.

Related Information

[Specifying General Properties of a Dispatcher \[page 388\]](#)

[Configuring Custom Parameters for a Dispatcher \[page 389\]](#)

[Suspending and Resuming a Dispatcher \[page 392\]](#)

[Stopping a Dispatcher \[page 393\]](#)

9.1.3.2 Policy (Dispatcher)

The [Policy](#) tab contains the performance optimization parameters.

The [Policy](#) tab of the dispatcher properties contains the following fields:

Field	Description
Check intv	Select a unit and enter a value to specify the interval between each time the dispatcher should check for scheduled jobs. The default value is 5 seconds. If more dispatchers are running on the same Identity Center, this number should be increased. Typically it should be set so that the average is 5 seconds, i.e. if there are three dispatchers, the default value should be multiplied by the number of dispatchers.
Handle tasks	Select this option to specify that this dispatcher should process tasks. The dispatcher is responsible for expanding and evaluating processes, conditional and switch tasks.
Evaluate relations	Select this option to specify that the dispatcher should recalculate relations (assignments) to check if there are event tasks that should be executed for any assignments. Assignment validity will also be handled.
Evaluate approvals	Select this option to specify that the dispatcher should process approvals. This involves all approval processing, including initiating an approval, sending notification messages and processing approve/decline events initiated from the Identity Management User Interface.
Housekeeping actions	Select Housekeeping actions to specify that the dispatcher should execute the scheduled procedures for housekeeping. The default value is that the dispatcher executes these procedures.  Note Make sure that at least one dispatcher executes these procedures.
Housekeeping intv	Select a unit and enter a value to specify the interval between each time the dispatcher should perform the defined housekeeping actions. The default value is 30 seconds.
Run jobs	Select Run jobs to specify that the dispatcher should start the runtime engine to run jobs.
Run provisioning jobs	Select this option to specify that the dispatcher should check for provisioning jobs for the specified runtime engine.
Run regular jobs	Select this option to specify that the dispatcher should check for regular jobs (i.e. ordinary Identity Center jobs, as opposed to jobs in the provisioning system) for the specified runtime engine.

Field	Description
Log level	Select a log level for the Java runtime engine. You can choose between: • None • Error • Info • Extended • Debug The default value is Info. When developing a solution or investigating a problem, increase the log level to include more information in the job log. In a production system, you can decrease the log level to avoid unnecessary information in the logs.
Stack trace	Select a stack trace level for the Java runtime engine. You can choose between: • None • Topmost entry • Full trace The default value is Topmost entry.
Java options	Enter any Java options to include on the command line for the runtime engine. This could for instance be -Xmx512m.

Related Information

[Configuring the Dispatcher Policy \[page 390\]](#)

9.1.3.3 Jobs (Dispatcher)

On the [Jobs](#) tab, the jobs which the dispatcher is responsible for can be specified.

The [Jobs](#) tab of the dispatcher properties contains the following fields:

Field	Description
Search Field	Search for jobs by providing the job's name.
Run Jobs	All jobs in the Identity Management are listed in a table, where you can select which jobs to be run by the dispatcher. You can sort jobs by clicking the column headers Identity Store Name, Package and Schedule Rule.

Related Information

[Specifying the Jobs of a Dispatcher \[page 391\]](#)

9.1.3.4 Advanced (Dispatcher)

In the [Advanced](#) tab you can configure custom parameters for the dispatcher.

The [Advanced](#) tab of the dispatcher properties contains the following fields:

Field	Description
Custom parameters list	The field is a list of parameter-value pairs. Note Do not change the parameters unless you are sure about the consequences. You can configure the following parameters: • DISP_MAX_JAVA_CONCURRENT_RUNTIMES_PERCENT Percentage of the total number of runtimes allocated for the Java runtime engine. Values: 1-100% (default 90). • DISP_MAX_WINDOWS_CONCURRENT_RUNTIMES_PERCENT Percentage of the total number of runtimes allocated for the Windows runtime engine. Values: 1-100% (default 50). • DISP_RUNTIME_COUNT_MODE Process count mode. Values: 0 1: • 0: PROCESS. The dispatcher keeps track of started processes (processids in the list). • 1: IDS. The dispatcher executes the stored procedures MC_LIST_RUNNING_ENGINES and MC_LIST_RUNNING_ENGINES_DISP in order to obtain the total number of running runtimes and the number of runtimes for a particular dispatcher. • DISP_RUNTIME_START_DELAY Delay between starting the multiple runtime engines in # of milliseconds. Values: 1-5000 (default 200). • DISP_THREAD_COUNT_REG Thread count regulator. Comma separated list, where the values are thresholds (in # of elements to be processed by dispatcher) for when the new threads are created. Values: #, #,..., N. Default value is 1000. E.g. value 300 means that if the total number of elements to process exceeds 300, the dispatcher will run with two threads (otherwise only with one). Value 300, 600 means that if the total number of elements to process • is lower than 300, there will only be one thread. • exceeds 300 but lower than 600, there will be two threads.

Field	Description
	- exceeds 600, there will be three threads. Usually, there should be no need to run more than three dispatcher threads. LOG_MODULES Dispatcher modules included in log. This corresponds to the policy selected for the dispatcher. Values: [ORDERED,] [UNORDERED,][CONDITIONAL,][SWITCH,][MXP_APPR,][EVAL_LINKS,] [EVAL_APPR,][JOBEXECUTE,][HOUSE_KEEP]. Default empty (logging from all modules). Note Normally you will use this if you need to debug the processing of the dispatcher. To avoid too many log entries, it will be useful to select only the modules of interest. MODE The mode of the runtime. May be used for debugging. Values: loop run. - loop: Checks for further jobs to process. - run: Stops after executing the job. MXDISPATCHER_CONFIGRELOAD_FREQUENCY Configuration reload frequency in # of seconds (default 30). MXDISPATCHER_DIE_CHECK_FREQUENCY Check for stop frequency in # of seconds (default 2). MXDISPATCHER_EXECSTRING STRING/STRING ARRAY when starting runtime. Values: 0 1. 0: The dispatcher builds a STRING ARRAY when starting the runtime engine. 1: The dispatcher builds a STRING when starting runtime engine. MXDISPATCHER_LOGTYPE Specifies where the dispatcher should send its logs. Values: 1 2. 1: Identity Center system log. 2: Console. MXDISPATCHER_REPORTEXEC Print runtime command. Values: 0 1. 0: Do not print the runtime command. 1: Print the whole command that is executed for starting runtime to STANDARD OUT. MXDISPATCHER_STREAM_PROCESSING Runtime stream processing. Controls how the dispatcher deals with standard out (and standard err) of the runtime. Values: 0 1. 0: Close. The connection to the runtime engine is closed, and logging from the runtime engine will not block the processing of the dispatcher. 1: Gobble. Error messages thrown by runtime are also logged by dispatcher. If not Closed or Gobbled, the runtime is blocked when buffer is full.

Related Information

[Configuring Custom Parameters for a Dispatcher \[page 389\]](#)

9.1.4 Entry Type Properties

Overview of the tabs of the entry type properties.

The entry type properties contain the following tabs:

General	Configure general parameters of the entry type (e.g. name, description) and access limitations.
Event Processes	Configure the event processes of the entry type.
Relations	Use the Relations tab to configure the relations for entries of this type.
Attributes	Use this tab to define which attributes are allowed and which are mandatory for the entry type.

Related Information

[General \(Entry Type\) \[page 981\]](#)

[Event Processes \(Entry Type\) \[page 983\]](#)

[Relations \(Entry Type\) \[page 984\]](#)

[Attributes \(Entry Type\) \[page 985\]](#)

9.1.4.1 General (Entry Type)

Gives an overview of the properties on the [General](#) tab of the entry type.

The [General](#) tab of the entry type properties contains the following fields:

Field	Description
Entry type ID	This field contains the entry type's unique identifier. This cannot be modified.
Name	Enter a name for the entry type.

Field	Description
	 Note Never let the entry type name start with "MX" or "SAP". You would typically make your own prefix to all user defined entry types.
<i>Display name</i>	Enter the name to be used when displaying the entry type in the Identity Management User Interface.
<i>Description</i>	Enter a text describing the entry type.
<i>Identity entry type</i>	This option indicates whether the entry type is used for license check. All entries with this entry type are counted as identity entries. For more information about licensing, see SAP Licenses and Global License Auditing Service in Related Links section.
<i>Context entry type</i>	Select this option to specify that this entry type is used for contexts. Entry types defined as context entry types are used when defining context types for roles, privileges and repositories. These entry types will also be used as available context types in the Identity Management User Interface.
<i>Display form</i>	Select a form that is used to display details about entries of this type in the Identity Management User Interface. The file <code>com.sap.idm.forms.default.idmpck</code> found in the folder <code><Install_folder>Core\ConfigurationPackages\Forms</code> contains a set of WebDynpro-based sample display forms for the entry types.  Note In SAP Identity Management User Interface, the maximum number of 20 multivalue entry reference attributes are displayed in the <i>Details about <entry></i> pane on the <i>Manage</i> tab. Attributes exceeding this limit will only be displayed if you open the Display form of the entry.
<i>Searchable entry type</i>	Select this option to specify that this entry type should be searchable from the Identity Management User Interface. This option is required in order to use the entry type with REST API version 2.
<i>Search form</i>	Select a form that is used to search for entries of this type in the Identity Management User Interface. The file <code>com.sap.idm.forms.default.idmpck</code> found in the folder <code><Install_folder>Core\ConfigurationPackages\Forms</code> contains a set of WebDynpro-based sample search forms for the entry types.
<i>Search attribute</i>	Select which attribute should be used as the search attribute for access limitations.

Field	Description
User attribute	Select which attribute should be used as the user attribute for access limitations.
Use access limitations for display task	By default, the display form will display information about any other entry (user) in the identity store, regardless of what is defined as access limitations. If access limitations is used only to limit search result, this is the way it should work. The display form should not display entries not matching the criteria defined with access limitations. Use this option to specify that the display form should use the criteria specified for access limitations.

Related Information

[SAP Licenses](#)

[System Measurement](#) 

[Entry Type Properties](#) [page 981]

[Event Processes \(Entry Type\)](#) [page 983]

[Relations \(Entry Type\)](#) [page 984]

[Attributes \(Entry Type\)](#) [page 985]

9.1.4.2 Event Processes (Entry Type)

Gives an overview of the properties on the [Event Processes](#) tab of the entry type.

The [Event Processes](#) tab of the entry type properties contains the following fields:

Field	Description
Add	Select the provisioning process of type Add Entry Process that is to be executed when an entry of this entry type is added to the identity store.
Modify	Select the provisioning process of type Modify Entry Process that is to be executed when one of the attributes of an entry of this entry type is modified.
Delete	Select the provisioning process of type Delete Entry Process that is to be executed when an entry of this entry type is deleted from the identity store.

 **Note**
 When this process is executed, the entry has already been removed.

Related Information

[Entry Type Properties \[page 981\]](#)

[General \(Entry Type\) \[page 981\]](#)

[Relations \(Entry Type\) \[page 984\]](#)

[Attributes \(Entry Type\) \[page 985\]](#)

9.1.4.3 Relations (Entry Type)

Gives an overview of the properties on the *Relations* tab of the entry type.

The *Relations* tab of the entry type properties contains the following fields:

Field	Description
<i>Allow subordinates</i>	Select this option to specify that entries of this entry type can have sub-entries. You can specify which entry types are allowed for the sub-entries in the list below.
<i>Allow multiple superiors of this entry type</i>	Select this option to specify that the entries can have more than one superior entry of this entry type. This setting is used to specify whether you want to create a strictly hierarchical tree (not allow multiple superiors) or if you want to create a mesh (allow multiple superiors). An example of the first case is locations. Here, you normally want a strict hierarchy, where one person is related to only one location. An example of the latter case is groups, where one person is normally a member of more than one group and a group can be a member of more than one other group.

Related Information

[Entry Type Properties \[page 981\]](#)

[General \(Entry Type\) \[page 981\]](#)

[Event Processes \(Entry Type\) \[page 983\]](#)

[Attributes \(Entry Type\) \[page 985\]](#)

9.1.4.4 Attributes (Entry Type)

Gives an overview of the properties on the [Attributes](#) tab of the entry type.

The [Attributes](#) tab of the entry type properties lists all attributes. Use the columns to configure the attributes for the entry type:

Field	Description
Allow	Select Allowed to specify that this attribute will be included in the list of attributes when configuring a task.
Mandatory	Select Mandatory to specify that this attribute is mandatory for the entry type. The attribute is always included when configuring the task attributes.
List	Select List to specify that the attribute is included in the search result table in the User Interface.
Ext. properties	For multivalue reference attributes, you can choose ... to open the Extended attribute properties dialog box where you can configure properties that are used for instance for role assignments.

Related Information

[Entry Type Properties \[page 981\]](#)

[General \(Entry Type\) \[page 981\]](#)

[Event Processes \(Entry Type\) \[page 983\]](#)

[Relations \(Entry Type\) \[page 984\]](#)

9.1.4.4.1 Extended Attribute Properties

Gives an overview of the properties on the [Extended attribute properties](#) dialog box.

The [Extended attribute properties](#) dialog box contains the following fields:

Field	Description
Reason	Specify how reason is displayed when viewing or modifying a multivalue reference attribute. You can choose between: • No Reason is not displayed. • Optional

Field	Description
	Reason is displayed, but is not required to fill in if you modify the assignment. Mandatory Reason is displayed and is required when modifying the assignment.
Validity	Select Validity to specify that the validity is included in the properties of the multivalue reference attribute.
Context	Select this option to specify that context is included in the assignment properties.  Note Only available for the MXREF_MX_ROLE and MXREF_MX_PRIVILEGE attributes.
Assignment status	Select this option to specify that the status of the assignment is included in the properties.  Note Only available for the MXREF_MX_ROLE and MXREF_MX_PRIVILEGE attributes.

Related Information

[Entry Types \(Attribute\) \[page 1001\]](#)

[Attribute Properties \[page 987\]](#)

[Attributes \(Entry Type\) \[page 985\]](#)

[Entry Type Properties \[page 981\]](#)

[Adding or Modifying an Identity Store Attribute \[page 173\]](#)

[Defining Entry Types for an Identity Store Attribute \[page 185\]](#)

[Configuring Extended Attribute Properties \[page 172\]](#)

9.1.5 Attribute Properties

Overview of the tabs of the attribute properties.

The attribute properties contain the following tabs:

<i>General</i>	Configure general parameters of the attribute (e.g. name, description) and attribute ownership.
<i>Storage</i>	Specify how the attribute is stored in the identity store.
<i>Presentation</i>	Specify how the attribute is presented in the Identity Management User Interface.
<i>Assignments</i>	For assignment attributes only: Configure the presentation of an assignment attribute in the Identity Management User Interface.
<i>Validation</i>	Configure input validation in the Identity Management User Interface.
<i>Event Processes</i>	Specify event processes for the attribute.
<i>Attribute Values</i>	Configure legal values for the attribute in the Identity Management User Interface.
<i>Entry Types</i>	Define in which entry types the attribute is included.

Related Information

[General \(Attribute\) \[page 988\]](#)

[Storage \(Attribute\) \[page 989\]](#)

[Presentation \(Attribute\) \[page 992\]](#)

[Assignments \(Attribute\) \[page 996\]](#)

[Validation \(Attribute\) \[page 998\]](#)

[Event Processes \(Attribute\) \[page 999\]](#)

[Attribute Values \(Attribute\) \[page 999\]](#)

[Entry Types \(Attribute\) \[page 1001\]](#)

[Adding or Modifying an Identity Store Attribute \[page 173\]](#)

[Configuring Attribute Ownership and Conflict Handling \[page 175\]](#)

[Defining the Storage Parameters of an Identity Store Attribute \[page 176\]](#)

[Configuring the Web Presentation of an Identity Store Attribute \[page 176\]](#)

[Configuring the Web Presentation of an Assignment Attribute \[page 177\]](#)

[Defining the Input Validation for an Identity Store Attribute \[page 178\]](#)

[Defining Event Processes for an Identity Store Attribute \[page 179\]](#)

[Defining Attribute Values for an Identity Store Attribute \[page 180\]](#)

9.1.5.1 General (Attribute)

Gives an overview of the properties on the *General* tab of the attribute.

The *General* tab of the attribute properties contains the following fields:

Field	Description
<i>Attribute ID</i>	The attribute's unique ID. The Identity Center suggests the next free number, but you can specify another (unused) number. This can be used to generate universal numbering for attributes, which may be useful in moving jobs between different Identity Center databases. This number cannot be changed later.
<i>Name</i>	Enter the name of the attribute. We recommend using the following characters: A-Z/a-z, 0-9 and _. The following characters are reserved and cannot be used in attribute names: • . Period • [] Square brackets • <> Brackets • {} Curly brackets • !! Two exclamation marks Note Attributes starting with MX_ are reserved for use by SAP. Never start a user defined attribute with MX_.
<i>Description</i>	Enter a text describing the attribute.
<i>Owner Repository</i>	Attribute ownership is used to determine which repositories are allowed to update the attribute in the identity store. To specify an owner for this attribute, select a repository in the list. If no owner should be specified for this attribute, select <i>None</i>. Note The Identity Management User Interface must be defined as a repository to be included in the list of available repositories.
<i>Conflict Handling</i>	Specify how precedence conflicts should be handled for this attribute. Such conflicts occur when a job that is not defined as owner repository tries to update an attribute already containing a value from the owner repository. Select one of the following values: • <i>Silent Ignore</i>

Field	Description
	Warning The job trying to write to the identity store generates a warning. Error The job trying to write to the identity store generates an error.
Log Conflicts to System Log	Select this option to specify that precedence conflicts are written to the system log.

Related Information

[Attribute Properties \[page 987\]](#)
[Storage \(Attribute\) \[page 989\]](#)
[Presentation \(Attribute\) \[page 992\]](#)
[Assignments \(Attribute\) \[page 996\]](#)
[Validation \(Attribute\) \[page 998\]](#)
[Event Processes \(Attribute\) \[page 999\]](#)
[Attribute Values \(Attribute\) \[page 999\]](#)
[Entry Types \(Attribute\) \[page 1001\]](#)
[Adding or Modifying an Identity Store Attribute \[page 173\]](#)
[Configuring Attribute Ownership and Conflict Handling \[page 175\]](#)

9.1.5.2 Storage (Attribute)

Gives an overview of the properties on the [Storage](#) tab of the attribute.

The [Storage](#) tab of the attribute properties contains the following fields:

Field	Description
Attribute Type	Select an attribute type for the attribute. There are some pre-defined attribute types. Some of them are used for specific purposes in the Identity Management User Interface.
Data Type	Select a data type for the attribute. You can choose between: Text Numeric Binary Datetime

Field	Description
	• Entry Reference Contains a reference to an MSKEY in the identity store. An error message is returned if the attribute does not contain a valid MSKEY. • Audit Flag Used for the attribute that contains audit flags. An error message is returned if the attribute does not contain a valid audit flag. • Task Reference Used to contain a reference to a task. An error message is returned if the attribute does not contain a valid task ID. • Attribute Reference Used to contain a reference to an identity store attribute. An error message is returned if the attribute does not contain a valid attribute name.
Allow Circular References	Select this option to specify that two entries are allowed to reference each other, for instance that two users can be defined as each other's substitutes. Note This option is only active if the selected data type is Entry Reference and the attribute name does not start with MXREF_, MXMEMBER_ and MX_AUTO. This means that circular references cannot be allowed for member relations, like roles and privileges.
Multivalue Attribute	Select this option to specify that this attribute can contain multiple values. The values are case sensitive, and each value must be unique. You can add or remove single values of a multivalue attribute.
Reference Entry Type	This is only enabled when the data type Entry Reference is selected. Select None to specify that any entry within the identity store can be used when creating the reference. To limit this to a specific entry type, select the entry type in the list. You can use this if you have a "Manager" attribute for instance, in which case the entry type should be set to Person.
Access Type	Select one of the access types in the list: • Read Write Compare • Compare Only Most attributes are of the type Read Write Compare. Compare Only is mostly used for passwords and other attributes that should not be read.
Encryption	Specify any encryption of the attribute. You can choose between: • None No encryption. (This is the default value.) • Encrypted • Hashed

Field	Description
	Attribute values are hashed using the default hashing algorithm. This (one-way) encryption method is used when the encrypted data is not going to be decrypted again, but rather used for comparison, as with the authentication password. An attribute containing the "answer" used for password recovery is typically encrypted using this encryption method.
<i>Time to Live</i>	An attribute's time to live specifies how long it will be valid (ValidTo). This means how long the attribute will be present in the identity store. After the given time, the attribute will automatically be deleted. Select a unit and enter a value to define how long the attribute should be present in the identity store. After the given time, the attribute value is deleted from the identity store and is moved to the historical values, if the number of revisions is set to 1 or higher. Choose <i>Indefinite</i> to specify that the attribute should never be removed from the identity store. To specify that the attribute should never be present in the identity store, select a unit and enter 0 as the value.
<i>Keep No. of Revisions/Days</i>	When an attribute is updated, the previous values can be kept in a separate table. This is used to keep a history of all changes to the attribute for each entry in the identity store. You can choose whether you want to keep a given number of revisions, or to keep the values for a given number of days. Select <i>Revisions</i> and specify how many revisions should be kept. If you enter 0, only the current value is stored. Select <i>Days</i> Attribute values are encrypted using the default encryption and specify the number of days for which you want to keep old values. The default value is 30. Enter 0 to specify that only the current value should be stored. To specify that all values should be kept indefinitely, select <i>Days</i> and enter -1. For more information, see Cleaning Up Historic Values in the Identity Store

Related Information

[Attribute Properties \[page 987\]](#)
[General \(Attribute\) \[page 988\]](#)
[Presentation \(Attribute\) \[page 992\]](#)
[Assignments \(Attribute\) \[page 996\]](#)
[Validation \(Attribute\) \[page 998\]](#)
[Event Processes \(Attribute\) \[page 999\]](#)
[Attribute Values \(Attribute\) \[page 999\]](#)
[Entry Types \(Attribute\) \[page 1001\]](#)
[Adding or Modifying an Identity Store Attribute \[page 173\]](#)

9.1.5.3 Presentation (Attribute)

Gives an overview of the properties on the *Presentation* tab of the attribute.

The *Presentation* tab of the attribute properties contains the following fields:

Field	Description
<i>Display Name</i>	Enter the name that is used when displaying the attribute in the Identity Management User Interface.
<i>Tooltip Text</i>	Enter a text that will appear as a tooltip for the field in the Identity Management User Interface.
<i>Presentation</i>	Specify what type of input field you want for this attribute. This must be seen in combination with other properties of the attribute. For instance if you have defined a list of available values for the attribute on the <i>Attribute Values</i> tab, you should select <i>SingleSelect</i> or <i>MultiSelect</i>. This property is passed to the Identity Management User Interface which interprets and processes the information before it is displayed. Possible values are: • <i>SingleLine</i> A normal input field (one line). • <i>MultiLine</i> An input field with several lines. • <i>SingleSelect</i> Drop-down list where the user can select a single value. → Recommendation Having a large list of predefined values is not recommended as this may slow down loading the form in the Identity Management User Interface or may lead to connection issues. • <i>MultiSelect</i> A list box where the user can select several values. • <i>Boolean</i> A check box. • <i>Picture</i> Not implemented in the Identity Management User Interface. • <i>Referral</i> Creates a reference to another entry in the identity store. A search interface is displayed, where the user can find the correct entry to create the reference. • <i>File</i>

Field	Description
	Displays a Browse... button that allows the user to find a file to attach.
	• Lookup Displays the description corresponding to a given ID. • Radio Button Displays a list of values as radio buttons. • Date Displays a date with the format YYYY-MM-DD. • Mail Displays the value as a mail address. • AutoContexts Displays a list of default contexts. • ObjectValueHelp Displays a list of multi-language values. • Process Offset When to execute a role/privilege member event task. • Repository Search for and select a repository. • Context Type Lists legal context type(s) for a role/privilege. • Context Entry Reference Displays all the entry types that are defined as context entry types. • Timeout Timeout for a master privilege. • Entry Visibility Entry visibility for a role, privilege or dynamic group. • Add Member Process Lists all public Add member processes. • Modify Member Process Lists all public Modify member processes. • Remove Member Process Lists all public Remove member processes. • Validate Add Process Lists all public Validate add processes. • Validate Modify Process Lists all public Validate modify processes. • Validate Remove Process Lists all public Validate remove processes. • Notify Process Lists all public Notify processes. • Result Handling OK Process Lists all public processes defined for Result handling as Execute process on OK result. • Result Handling Fail Process

Field	Description
	Lists all public processes defined for Result handling as Execute process on Failed result. • Provisioning Lists all public Provisioning processes. • Deprovisioning Lists all public Deprovisioning processes. • Privilege Modify Process Modify process for a privilege. • Missing Master Privilege Process Missing master privilege process for a privilege.
	Note Only the following presentation types are supported for the search forms: • SingleSelect (Singlevalue Attribute) • Boolean (Singlevalue Attribute) • SingleLine (Singlevalue Attribute) • MultiLine (Singlevalue Attribute) • Date (Singlevalue Attribute) • Process Offset (Singlevalue Attribute) • Entry visibility (Multivalue Attribute) • Timeout (Singlevalue Attribute) • Result Handling OK Process (Multivalue Attribute)
	Note In SAP Identity Management User Interface, filtering attribute values (defined on the Attribute Values tab) depends on the combination of the attribute data type, presentation and number of values it can contain (single- or multivalue attribute). Some combinations of attribute properties make filtering of attribute values impossible. For example: single-value attribute of Data Type Text and Presentation Repository, such as MX_REPOSITORYNAME.
Confirm Input	Select this option to specify that the user has to confirm the value entered in this field by re-entering it in a separate field.
Hide Input	Select this option to specify that the input should be hidden (replaced by dots). This is typically used for passwords or other sensitive information.
Show Search Field	Select this option to specify that the presentation of this attribute will include a possibility to search for values in the list. This is particularly useful if there may be many values assigned to a multivalue attribute. If this is selected, no values will be shown in the user interface, until a search is done.

Field	Description
	Note This is only enabled for multivalue attributes.
<i>List Entries on Load</i>	Note This option is only editable for multivalue attributes. Select this option to specify that existing attribute values are listed when opening the task. This can be disabled for performance reasons if there is a large number of values to load.
<i>Number of Lines</i>	This field is used for two purposes: - For multivalue attributes: Enter the number of rows (values) you want to list for multivalue attributes. - For multiline fields: Enter the number of lines to display for an attribute displayed in a multiline field. The default value is 5.
<i>Read Only</i>	Select <i>Read Only</i> to specify that the default value for this attribute will be set to read only when configuring the attributes for a task.
<i>Mandatory</i>	Select <i>Mandatory</i> to specify that the default value for this attribute will be set to mandatory when configuring the attributes for a task.

Related Information

[Attribute Properties \[page 987\]](#)

[General \(Attribute\) \[page 988\]](#)

[Storage \(Attribute\) \[page 989\]](#)

[Assignments \(Attribute\) \[page 996\]](#)

[Validation \(Attribute\) \[page 998\]](#)

[Event Processes \(Attribute\) \[page 999\]](#)

[Attribute Values \(Attribute\) \[page 999\]](#)

[Entry Types \(Attribute\) \[page 1001\]](#)

[Adding or Modifying an Identity Store Attribute \[page 173\]](#)

[Configuring the Web Presentation of an Identity Store Attribute \[page 176\]](#)

9.1.5.4 Assignments (Attribute)

Available for assignment attributes only. Gives an overview of the properties on the [Assignments](#) tab of the attribute.

The [Assignments](#) tab of the assignment attribute properties contains the following fields:

Field	Description
Entry Type	The values you specify for the attribute are used as defaults for the attribute when configuring the attributes for a task. A typical use for this setting, is to only allow the user to add new roles, not privileges. But you will still be able to view the entire hierarchy of roles and privileges. Select which entry type(s) should be included in the Available list in the User Interface. You can choose between: • Privilege/Role Both privileges and roles can be added. The user can select which type of entries to show. The default entry type in the User Interface is role. • Privilege Only privileges can be added. • Role Only roles can be added. The default value is Privilege/Role.
Default Search Filter	The values you specify for the attribute are used as defaults for the attribute when configuring the attributes for a task. Enter a text that will be used as a filter in the Assigned list when displaying the task. The text is used as a case insensitive free text search. Only entries matching the text are displayed. If Read Only Search Filter is selected, the user is not allowed to change the filter.
Read Only Search Filter	The values you specify for the attribute are used as defaults for the attribute when configuring the attributes for a task. Select this option to specify that the user is not allowed to change any search filter entered as Default Search Filter.
Show Advanced Search	The values you specify for the attribute are used as defaults for the attribute when configuring the attributes for a task. The purpose is to simplify the User Interface, and limit the possibility of seeing future values or unassigned assignments. Deselect this option to specify that the user should not be able to use the advanced search in the Assigned list.
Include Direct Assignments Only	The values you specify for the attribute are used as defaults for the attribute when configuring the attributes for a task.

Field	Description
	Select <i>Include Direct Assignments Only</i> to specify that only direct assignments will be shown when searching. If a displayed role has inherited assignments, you will still be able to browse the hierarchy below the role. Just note that the displayed status of inherited assignments below the role is not up-to-date.
<i>Status Filter</i>	The values you specify for the attribute are used as defaults for the attribute when configuring the attributes for a task. You can select which assignments are displayed in the list based on the assignment's status. You can choose between the following values: • <i>All</i> All assignment are included in the list, regardless of status. • <i>Assigned</i> Only current assignments are included in the list. • <i>Not Assigned</i> Only assignments that are requested, but not assigned are included in the list. This means assignments that are requested, but either failed or rejected are displayed. Also assignments waiting for an approval are displayed.

Related Information

[Attribute Properties \[page 987\]](#)

[General \(Attribute\) \[page 988\]](#)

[Storage \(Attribute\) \[page 989\]](#)

[Presentation \(Attribute\) \[page 992\]](#)

[Validation \(Attribute\) \[page 998\]](#)

[Event Processes \(Attribute\) \[page 999\]](#)

[Attribute Values \(Attribute\) \[page 999\]](#)

[Entry Types \(Attribute\) \[page 1001\]](#)

[Adding or Modifying an Identity Store Attribute \[page 173\]](#)

[Configuring the Web Presentation of an Assignment Attribute \[page 177\]](#)

9.1.5.5 Validation (Attribute)

Gives an overview of the properties on the [Validation](#) tab of the attribute.

The [Validation](#) tab of the attribute properties contains the following fields:

Field	Description
Mixed Case Characters	Select this validation option to specify that the value must contain both upper and lower case characters.
Mixed Letters and Numbers	Select this validation option to specify that the value must contain both letters and numbers.
Unique Value	Select Unique Value to specify that the attribute must contain a unique value. If a duplicate value is entered, the operation is canceled and any changes rolled back. An error message is written to the job log.
Minimum Length	This parameter is used to verify the values entered for this attribute in the Identity Management User Interface. Enter the minimum length that should be allowed for this attribute. Value 0 means no check.
Maximum Length	This parameter is used to verify the values entered for this attribute in the Identity Management User Interface. Enter the maximum length that should be allowed for this attribute. Value 0 means no check.
Regex	Enter a regular expression that is used to validate a value entered in the field.

Related Information

[Regex - regular expressions](#) 

[Attribute Properties \[page 987\]](#)

[General \(Attribute\) \[page 988\]](#)

[Storage \(Attribute\) \[page 989\]](#)

[Presentation \(Attribute\) \[page 992\]](#)

[Assignments \(Attribute\) \[page 996\]](#)

[Event Processes \(Attribute\) \[page 999\]](#)

[Attribute Values \(Attribute\) \[page 999\]](#)

[Entry Types \(Attribute\) \[page 1001\]](#)

[Adding or Modifying an Identity Store Attribute \[page 173\]](#)

[Defining the Input Validation for an Identity Store Attribute \[page 178\]](#)

9.1.5.6 Event Processes (Attribute)

Gives an overview of the properties on the *Event Processes* tab of the attribute.

The *Event Processes* tab of the attribute properties contains the following fields:

Field	Description
<i>Add</i>	Select the provisioning process of type <i>Add Attribute Process</i> that is to be executed when the attribute is added to an entry.
<i>Modify</i>	Select the provisioning process of type <i>Modify Attribute Process</i> that is to be executed when an attribute value is modified.
<i>Delete</i>	Select the provisioning process of type <i>Delete Attribute Process</i> that is to be executed when the attribute is deleted from an entry. If you add a delete task to the MSKEYVALUE attribute, this will be executed when the entire entry is deleted from the identity store.

Related Information

[Attribute Properties \[page 987\]](#)

[General \(Attribute\) \[page 988\]](#)

[Storage \(Attribute\) \[page 989\]](#)

[Presentation \(Attribute\) \[page 992\]](#)

[Assignments \(Attribute\) \[page 996\]](#)

[Validation \(Attribute\) \[page 998\]](#)

[Attribute Values \(Attribute\) \[page 999\]](#)

[Entry Types \(Attribute\) \[page 1001\]](#)

[Adding or Modifying an Identity Store Attribute \[page 173\]](#)

[Defining Event Processes for an Identity Store Attribute \[page 179\]](#)

9.1.5.7 Attribute Values (Attribute)

Gives an overview of the properties on the *Attribute Values* tab of the attribute.

The *Attribute Values* tab of the attribute properties contains the following fields:

Field	Description
<i>Value</i>	Select to define a list with legal values for the attribute. Choose the following:

Field	Description
	• Add to add a new legal value to the list. • Modify to modify the list of legal values. • Delete to delete a legal value from the list.
SQL query	Select SQL query and enter a query that is used to retrieve the legal values for the attribute.
Value help	Select Value help and enter a reference to a table containing the value help.
Table name	Enter the name of the table containing the value help. The table that is installed with the product is mxi_AttrValueHelp.
Values ID	Enter the ID for the values that should be displayed for this attribute (Valid).
Language dependent	Select this option to specify that the values for this attribute are language dependent.
Task	Select this option to specify a task for which the access control is used to retrieve the legal values.

Note

In SAP Identity Management User Interface, filtering attribute values (defined on the [Attribute Values](#) tab) depends on the combination of the attribute data type, presentation and number of values it can contain (single- or multivalue attribute).

Some combinations of attribute properties make filtering of attribute values impossible. For example: single-value attribute of [Data Type](#) Text and [Presentation](#) Repository, such as MX_REPOSITORYNAME.

Note

Only in the standard SAP Identity Management User Interface all of the configured options are supported. For the REST API, ToldentityStore pass, and VDS only the defined legal values in [Value list](#) are validated against the provided value.

Related Information

[Attribute Properties \[page 987\]](#)

[General \(Attribute\) \[page 988\]](#)

[Storage \(Attribute\) \[page 989\]](#)

[Presentation \(Attribute\) \[page 992\]](#)

[Assignments \(Attribute\) \[page 996\]](#)

[Validation \(Attribute\) \[page 998\]](#)

[Event Processes \(Attribute\) \[page 999\]](#)

[Entry Types \(Attribute\) \[page 1001\]](#)

[Adding or Modifying an Identity Store Attribute \[page 173\]](#)
[Defining Attribute Values for an Identity Store Attribute \[page 180\]](#)
[Defining an SQL Query to Create a List of Legal Values \[page 180\]](#)
[Defining an SQL Query to Create a Value Dependent List of Legal Values \[page 182\]](#)
[Defining Attribute Value Help \[page 183\]](#)

9.1.5.8 Entry Types (Attribute)

Gives an overview of the properties on the *Entry types* tab of the attribute.

The *Entry types* tab of the attribute properties contains the following fields:

Field	Description
<i>Allow</i>	Select <i>Allow</i> to specify that this attribute will be included in the list of attributes when configuring a task.
<i>Mandatory</i>	Select <i>Mandatory</i> to specify that this attribute is mandatory for the entry type. The attribute is always included when configuring the task attributes.
<i>List</i>	Select <i>List</i> to specify that the attribute is included in the search result table in the User Interface.
<i>Ext. properties</i>	For multivalued reference attributes, you can choose ... to open the <i>Extended attribute properties</i> dialog box where you can configure properties that are used for role assignments for instance.

Related Information

[Attribute Properties \[page 987\]](#)
[General \(Attribute\) \[page 988\]](#)
[Storage \(Attribute\) \[page 989\]](#)
[Presentation \(Attribute\) \[page 992\]](#)
[Assignments \(Attribute\) \[page 996\]](#)
[Validation \(Attribute\) \[page 998\]](#)
[Event Processes \(Attribute\) \[page 999\]](#)
[Attribute Values \(Attribute\) \[page 999\]](#)
[Adding or Modifying an Identity Store Attribute \[page 173\]](#)
[Defining Entry Types for an Identity Store Attribute \[page 185\]](#)
[Configuring Extended Attribute Properties \[page 172\]](#)
[Extended Attribute Properties \[page 985\]](#)

9.1.5.8.1 Extended Attribute Properties

Gives an overview of the properties on the [Extended attribute properties](#) dialog box.

The [Extended attribute properties](#) dialog box contains the following fields:

Field	Description
Reason	Specify how reason is displayed when viewing or modifying a multivalue reference attribute. You can choose between: • No Reason is not displayed. • Optional Reason is displayed, but is not required to fill in if you modify the assignment. • Mandatory Reason is displayed and is required when modifying the assignment.
Validity	Select Validity to specify that the validity is included in the properties of the multivalue reference attribute.
Context	Select this option to specify that context is included in the assignment properties. Note Only available for the MXREF_MX_ROLE and MXREF_MX_PRIVILEGE attributes.
Assignment status	Select this option to specify that the status of the assignment is included in the properties. Note Only available for the MXREF_MX_ROLE and MXREF_MX_PRIVILEGE attributes.

Related Information

[Entry Types \(Attribute\) \[page 1001\]](#)

[Attribute Properties \[page 987\]](#)

[Attributes \(Entry Type\) \[page 985\]](#)

[Entry Type Properties \[page 981\]](#)

[Adding or Modifying an Identity Store Attribute \[page 173\]](#)

[Defining Entry Types for an Identity Store Attribute \[page 185\]](#)

[Configuring Extended Attribute Properties \[page 172\]](#)

9.1.6 Package Folder Overview

The package overview provides information about all packages in the folder.

Use the field above the list as a filter if the list is long.

Column	Description
<i>Package Name</i>	The name that was specified when the package was created.
<i>Version</i>	The package's version number. This number consists of two parts, the major number and the minor number. The minor number is incremented every time the package is checked in after editing. The major number is incremented every time there are changes to the public objects of the package.
<i>Locked by</i>	The user that has checked out this package.
<i>Last Modified</i>	The date and time the package was last modified.
<i>Description</i>	A text describing the package.

9.1.7 Package Properties

Overview of the tabs of the package properties.

The package properties contain the following tabs:

<i>General</i>	View the general parameters of a package, like ID and name and who has locked the package.
<i>Public Objects</i>	View objects defined as public in the package.
<i>Dependencies</i>	View any external dependencies to this package.
<i>Authorized Users</i>	Maintain the list of authorized users of this package.
<i>Revision History</i>	View the package's revision history.

Related Information

[General \(Packages\) \[page 1004\]](#)

[Public Objects \(Packages\) \[page 1005\]](#)

[Dependencies \(Packages\) \[page 1005\]](#)

[Authorized Users \(Packages\) \[page 1009\]](#)

[Revision History \(Packages\) \[page 1010\]](#)

[Viewing and Configuring Package Properties \[page 191\]](#)

9.1.7.1 General (Packages)

Gives an overview of the properties on the *General* tab of the package properties.

The *General* tab of the package properties contains the following fields:

Field	Description
<i>Package ID</i>	The package's unique ID. The package is given a unique ID when it is created. This value cannot be changed.
<i>Package Name</i>	The name that was specified when the package was created. This name cannot be changed.
<i>Locked By</i>	If the package is checked out, the user's name is shown here.
<i>Locked Date/Time</i>	The date and time when the package was locked for editing.
<i>Version</i>	The package's version number. This number consists of two parts, the major number and the minor number. The minor number is incremented every time the package is checked in after editing. The major number is incremented every time there are changes to the public objects of the package.
<i>Package Type</i>	There are two types of packages: • <i>Customer Package</i> This is a normal package. • <i>SAP Package</i> A package distributed by SAP that cannot be modified.
<i>Last Modified</i>	The date and time the package was last modified.
<i>Description</i>	A text describing the package.

Related Information

[Viewing General Properties of a Package \[page 192\]](#)

9.1.7.2 Public Objects (Packages)

Gives an overview of the properties on the [Public Objects](#) tab of the package properties.

The [Public Objects](#) tab contains a table listing all public objects in the package. Public objects are processes or scripts that can be accessed by other packages, as well as forms that can be defined on entry types.

Use the field above the list to limit the number of objects to the ones of interest.

The table contains the following columns:

Column	Description
Exposed Name	The object name.
Object Type	The type of object (for example, process, script, form).
Is Referenced	A green check mark () icon indicates that the object is referenced by another package, while a red () icon indicates that the object is not referenced by another package.

Related Information

[Viewing Public Objects in a Package \[page 193\]](#)

9.1.7.3 Dependencies (Packages)

Gives an overview of the properties on the [Dependencies](#) tab of the package properties.

The [Dependencies](#) tab contains two tables listing all external dependencies for this package.

Use the field above the list to limit the number of objects to the ones of interest.

Outgoing

This table shows all references from this package to objects in other packages.

Column	Description
Status	The status of the dependency: • Linked OK • Linked to different version

Column	Description
	Objects in the package contain references to different package version. • Missing target The referenced object is missing. • Invalid repository ID
Package	The package of the referenced object. If the referenced object is a repository, the value is [Repository].
Object	The name of the object containing the reference. The symbol preceding the name indicates the type of object.
Reference	The referenced object. The symbol preceding the name indicates the type of object.
Required Version	This number indicates the required major and minor version of the referenced package. This number is maintained when you check in new versions of the referenced package. The major number is incremented every time the public interface is changed (for example, a process is made public). The minor number is incremented every time the package is checked in, regardless of whether the package has been modified. Adding cross-package references to public objects increments the minor version of the package.
Usage Type	The type of usage the referenced object is used for. • Links - The referenced object is a linked process or script. • Result Handling OK - The referenced object is a process that is used on OK Result (Task Result Actions). • Result Handling Failed - The referenced object is a process that is used on Failed Result (Task Result Actions). • Result Handling OK Chain - The referenced object is a process that is used on OK Result (Chain Result Actions). • Result Handling Failed Chain - The referenced object is a process that is used on Failed Result (Chain Result Actions). • Initialize - The referenced object is a process that is used on Initialize. • Notification - The referenced object is a process that is used for Notification. • Exception - The referenced object is a process that is used for Exception. • Add Attribute - The referenced object is a process used for adding an attribute. • Modify Attribute - The referenced object is a process used for modifying an attribute. • Delete Attribute - The referenced object is a process used for deleting an attribute.

Column	Description
	• Add Entry - The referenced object is a process used for adding an entry type. • Modify Entry - The referenced object is a process used for modifying an entry type. • Delete Entry - The referenced object is a process used for deleting an entry type. • Search - The referenced object is a search form defined for an entry type and used for searching entries of this entry type. • Display - The referenced object is a display form defined for an entry type and used for displaying entries of this entry type. • Process Reference Constant - The referenced object is a process. This process is the value of a package constant. • Repository Type Constant - The referenced object is a process. This process is the value of a repository constant (repository constant, repository type constant or repository type constant with override) that is defined on a repository type.
	Note Unlike job and repository constants of type Process Reference, job variables of type Process Reference are not displayed on the Dependencies tab of the package.

Incoming

This table shows all references from other packages to objects in this package.

Column	Description
Status	The status of the dependency: • Linked OK • Linked to different version Objects in the package contain references to different package version. • Target already occupied An event process or form is already defined on an entry type or attribute.
Package	The package of the referenced object.
Object	The name of the object containing the reference. The symbol preceding the name indicates the type of object.
Reference	The referenced object.

Column	Description
	The symbol preceding the name indicates the type of object.
<i>Required Version</i>	This number indicates the required major and minor version of the referenced package. This number is maintained when you check in new versions of the referenced package. The major number is incremented every time the public interface is changed (for example, a process is made public). The minor number is incremented every time the package is checked in, regardless of whether the package has been modified. Adding cross-package references to public objects increments the minor version of the package.
<i>Usage Type</i>	The type of usage the referenced object is used for. • <i>Links</i> - The referenced object is a linked process or script. • <i>Result Handling OK</i> - The referenced object is a process that is used on OK Result (Task Result Actions). • <i>Result Handling Failed</i> - The referenced object is a process that is used on Failed Result (Task Result Actions). • <i>Result Handling OK Chain</i> - The referenced object is a process that is used on OK Result (Chain Result Actions). • <i>Result Handling Failed Chain</i> - The referenced object is a process that is used on Failed Result (Chain Result Actions). • <i>Initialize</i> - The referenced object is a process that is used on Initialize. • <i>Notification</i> - The referenced object is a process that is used for Notification. • <i>Exception</i> - The referenced object is a process that is used for Exception. • <i>Add Attribute</i> - The referenced object is a process used for adding an attribute. • <i>Modify Attribute</i> - The referenced object is a process used for modifying an attribute. • <i>Delete Attribute</i> - The referenced object is a process used for deleting an attribute. • <i>Add Entry</i> - The referenced object is a process used for adding an entry type. • <i>Modify Entry</i> - The referenced object is a process used for modifying an entry type. • <i>Delete Entry</i> - The referenced object is a process used for deleting an entry type. • <i>Search</i> - The referenced object is a search form defined for an entry type and used for searching entries of this entry type. • <i>Display</i> - The referenced object is a display form defined for an entry type and used for displaying entries of this entry type. • <i>Process Reference Constant</i> - The referenced object is a process. This process is the value of a package constant.

Column	Description
	Repository Type Constant - The referenced object is a process. This process is the value of a repository constant (repository constant, repository type constant or repository type constant with override) that is defined on a repository type.

Related Information

[Viewing External Dependencies of a Package \[page 193\]](#)

9.1.7.4 Authorized Users (Packages)

Gives an overview of the properties on the [Authorized Users](#) tab of the package properties.

The [Authorized Users](#) tab shows a list of all users that have access to this package. Each user can have one or more roles.

Use the field above the list as a filter if the list is long.

Column	Description
User	The user's name. You can add or remove users and give access rights if you are the package owner.
View	The selected users are able to view the configuration, but not change anything.
Developer	The selected users are able to modify the configuration in this package.
Layout Developer	The selected users are able to modify the forms for the Identity Management User Interface.
Import	The selected users are allowed to import new versions of this package.
Owner	The selected users are defined as owners of this package.

Related Information

[Viewing Authorized Users of a Package \[page 195\]](#)

[Providing Authorizations to a Package \[page 196\]](#)

[Removing Authorization to a Package \[page 197\]](#)

9.1.7.5 Revision History (Packages)

Gives an overview of the properties on the [Revision History](#) tab of the package properties.

The [Revision History](#) tab provides an overview of all logged changes to this package.

Use the field above the list as a filter if the list is long.

Column	Description
Version	The package version. The number is maintained automatically. The major number is incremented every time the public interface is changed (for example, a process is made public). The minor number is incremented every time the package is checked in, regardless of whether the package has been modified. Adding cross-package references to public objects increments the minor version of the package.
Date	The date and time of the logged change.
User	The user that did the change. If it is a change done by the system, it will say SYSTEM .
Action	The type of change that was done. Possible values are: • Authorization granted • Authorization revoked • Package checked in • Package checked out • Package created • Package imported • Package restored
Description	The description of the action. This can either be the a text provided by the user, or a default text added by the system.
Import User	The user that imported the version.
Import Date	The date the version was imported.
Import Description	The description that was added when the package was imported.

Related Information

[Viewing Revision History of a Package \[page 198\]](#)

[Restoring Version of a Package \[page 198\]](#)

9.1.8 Constants (Package)

The Package Constant dialog box contains the following fields:

Field	Description
<i>Name</i>	Enter the name that you use when referencing the constant. Legal values are A-Z, 0-9 and _.
<i>Data Type</i>	Select a data type for the constant: • Boolean • Entry Reference • File Name • Folder Name • JDBC Driver • JDBC URL • Job Reference • LDAP URL • Password • Process Reference • String
<i>Value</i>	Enter the value of the constant.
<i>Encrypt Value</i>	Select this check box to specify that the value of the constant should be encrypted. It will then be hidden in the dialog box, and will be stored encrypted in the Identity Management database. Note You can only encrypt values of the following data types: String, Password, JDBC URL, LDAP URL and OLE DB Connection String (valid for repository type constants only). Note If you encrypt a value, you will NOT be able turn it off again later. You need to delete the value, close the dialog box, reopen it and enter a new value.
<i>Description</i>	Enter a text describing the constant.
<i>Show in the Admin User Interface</i>	Select this checkbox to make the package constant editable in SAP Identity Management Administration User Interface.

Related Information

[Adding Package Constants \[page 208\]](#)

9.1.9 Repository Type Properties

Overview of the tabs of the repository type properties.

The repository type properties contain the following tabs:

<i>General</i>	View the general information of the repository type and configure constants.
<i>Privileges</i>	Configure privilege policy settings.
<i>Assignment grouping</i>	Configure assignment grouping.
<i>Event process</i>	Define privilege member event processes.
<i>Context</i>	Configure contexts - define context types and autoassign context policy.

Related Information

[General \(Repository Type\) \[page 1012\]](#)

[Privileges \(Repository Type\) \[page 1014\]](#)

[Assignment Grouping \(Repository Type\) \[page 1015\]](#)

[Event Process \(Repository Type\) \[page 1016\]](#)

[Context \(Repository Type\) \[page 1021\]](#)

9.1.9.1 General (Repository Type)

Gives an overview of the properties on the *General* tab of the repository type.

The *General* tab of the repository type properties contains the following fields:

Field	Description
<i>Repository type ID</i>	The repository type's unique ID. The Developer Studio suggests the next free number (i.e. this ID is system generated) and this number cannot be changed.

Field	Description
<i>Repository type name</i>	Enter the name of the repository type. Avoid using special characters and space. We recommend using the following characters: A-Z/a-z, 0-9 and _ (i.e. the repository type name should be legal as a database table name).
<i>Belongs to package</i>	Shows the reference to the package where the repository type is created.

Constants

There are the following columns for each constant in the list:

- **Name:** Name of the constant.
- **Category:** Defines the category of the constant. There are three types of constants that can be defined on the repository type:
 - **Repository constant**
Repository constant specific for each repository of this type. For example: host name, login credentials, port number. It must be defined on each repository.
It is always displayed in the Identity Management Administration User Interface.
 - **Repository type constant**
Repository constant common to all repositories of this type. It is repository type specific. It is only defined on the repository type and cannot be overridden. For example: process references (event processes), job references.
It is never displayed in the Identity Management Administration User Interface.
 - **Repository type constant with override**
Repository constant common to all repositories of this type but may be overridden on a given repository. For example: grouping algorithm, port number and other behavior settings. By default, a repository type constant with override has the same value for all repositories of this type, but it can be modified on each repository.
It is always displayed in the Identity Management Administration User Interface.

📌 Note

All repository constants declared in a repository type are stored in `mc_repository_type_vars` database table. Repository constants defined for a specific repository are stored in the `mc_repository_vars` database table only.

Constant	VarCategory Database Column
Repository constant	-1
Repository type constant	-4

Field	Description				
	<table> <tr> <th>Constant</th><th>VarCategory Database Column</th></tr> <tr> <td>Repository type constant with override</td><td>-3</td></tr> </table> • Data Type: Defines the data type of the constant (for example, String, Boolean). • Value: Value of the given constant (can be encrypted). • Description: Description of the constant (if necessary). The constants can be added to the list (use the New button), modified (the Modify button), removed (the Delete button), and their order changed (use the buttons Up and Down to move the constants up/down the list). Note The order of the constants in this list will determine the order in which the constants are presented in repository of this repository type.	Constant	VarCategory Database Column	Repository type constant with override	-3
Constant	VarCategory Database Column				
Repository type constant with override	-3				

Related Information

[Defining the Constants for the Repository Type \[page 213\]](#)

[Repository Type Properties \[page 1012\]](#)

[Repository Types, Constants and Variables \[page 101\]](#)

9.1.9.2 Privileges (Repository Type)

Gives an overview of the properties on the [Privileges](#) tab of the repository type.

Use the [Privileges](#) tab to configure privilege policy settings. The [Privileges](#) tab of the repository type properties contains the following fields:

Field	Description
Missing	This policy setting is used when assigning a privilege and the master privilege is not (yet) assigned. Note The only option currently available is Wait.

Field	Description
	<i>Wait</i> means that the pending value object for the privilege is created, and the process is waiting for the master privilege to be assigned. As soon as the master privilege is created, execution of the pending value object process will start. If the master privilege is already defined, execution continues immediately.  Note There is no automatic assignment of the master privilege, but you can use the <i>No Master Process</i> to assign the privilege.
<i>No Master Process</i>	Select the process that is executed when the policy is <i>Wait</i> and the master privilege is not present. This process will then be executed when a privilege is assigned that requires the presence of the master privilege. This process can be used to assign the master privilege, by assigning the privilege directly or by assigning a role that references the privilege. When the master privilege is assigned, any assignments waiting for the master privilege will also be assigned.  Note There is no automatic removal of a master privilege assigned with the <i>No Master Process</i> if all depending privileges are removed from an entry.

Related Information

[Configuring Privilege Policy for a Repository Type \[page 214\]](#)
[Repository Type Properties \[page 1012\]](#)

9.1.9.3 Assignment Grouping (Repository Type)

Gives an overview of the properties on the *Assignment grouping* tab of the repository type.

Use the *Assignment grouping* tab to configure grouping of assignments. Grouping applies to all privileges which are assigned in the same operation. This can for example be a task executed from the user interface that assigns 3 roles, which, in turn, inherits a number of privileges.

Note

There is no handling of master privilege when removing privileges.

The [Assignment grouping](#) tab of the repository type properties contains the following fields:

Field	Description
No grouping	The member event tasks are executed for every privilege. This is the default value.
Grouping	The member event tasks are executed for a group of assignments. Assignments are separated according to what is selected below (Separate by operation , Separate by application or Separate by privilege attribute). If none of the check boxes are selected, all assignments will be handled by one execution of the member event tasks. The check boxes can be combined.
Separate by operation	Assignments are grouped by operation, i.e. add or remove operations are grouped and processed with one task for each operation.
Separate by application	Assignments are grouped according to application, i.e. all assignments to the same application are processed with one task.  Note The MX_APPLICATION_ID attribute on the privilege is used to determine the application.
Separate by privilege attribute	Assignments are grouped by the value of the specified privilege attribute. Assignments with the same value for the given attribute are processed with one task.

Related Information

[Configuring the Assignment Grouping \[page 214\]](#)

[Assignment Grouping \[page 57\]](#)

[Repository Type Properties \[page 1012\]](#)

9.1.9.4 Event Process (Repository Type)

Gives an overview of the properties on the [Event process](#) tab of the repository type.

The [Event process](#) tab of the repository type properties contains the following fields:

Field	Description
Validate Add	Select the process to verify that the add operation is allowed for the given entry. You can select a process of type Regular or Validate Add Process from the current package or any other package in the same identity store, if the process is made public. This can for

Field	Description
	example be an approval or a call to GRC Access Control. Use the ... button to navigate to and add the process. This process returns either true or false. If it contains an <i>Approval task</i>, this is handled automatically, but if you are using a regular task, for instance to check a group membership in an external system, this must be done manually by setting <code>MX_ATTR_STATE</code> to 1 to allow the assignment and 3 to reject it. Note If this process is defined, the <i>Add Member Process</i> will not be performed unless this process completes successfully. Select or fill in a value in the <i>Execute</i> field to specify when the process should be executed relative to the <i>valid from</i> and <i>valid to</i> specified for the assignment. You have the following options: - When valid: Execute the process at the <i>valid from</i> specified in the assignment request. - Immediately: Execute the process when the assignment is requested. This is the default value for validate processes. Note This option works only for direct assignments of privileges. It does not work for privileges assigned through roles. Assignments of roles containing privileges are made as soon as the <i>valid from</i> date is reached, then the role and the privilege will be assigned. - Number of days: In the <i>Execute</i> field, fill in a number of days before the <i>valid from</i> when the process should be executed. The <i>days before</i> description is displayed. For instance, if you enter <i>5</i>, the process will be executed 5 days before the date specified as <i>valid from</i> in the assignment request.
<i>Validate Remove</i>	Select the process that should be used to verify that the remove operation is allowed for the given entry. You can select a process of type <i>Regular</i> or <i>Validate Remove Process</i> from the current package or any other package in the same identity store, if the process is made public. This can for example be an approval or a call to GRC Access Control. Use the ... button to navigate to and add the process. The <i>Validate Remove Process</i> is only executed when the <i>Remove Member Process</i> is executed as result of an explicit deassignment, and not when the <i>Remove Member Process</i> is run when an assignment expires (by <i>valid to</i>). This process returns either true or false. If it contains an <i>Approval task</i>, this is handled automatically, but if you are using a regular task, for instance to check a group membership in an external system, this must be done manually by setting <code>MX_ATTR_STATE</code> to 1 to allow removing the assignment and 3 to reject it.

Field	Description
	 Note If this process is defined, the Remove Member Process will not be performed unless this process completes successfully.
Validate Modify Validity	Select a process that should be used to verify that the modify validity operation is allowed for the given entry. You can select a process of type Regular or Validate Modify Process from the current package or any other package in the same identity store, if the process is made public. This can for example be an approval or a call to GRC Access Control. Use the ... button to navigate to and add the process. This process returns either true or false. If it contains an Approval task, this is handled automatically, but if you are using a regular task, for instance to check a group membership in an external system, this must be done manually by setting MX_ATTR_STATE to 1 to allow modifying the assignment and 3 to reject it.  Note If this process is defined, the Modify Member Process will not be performed unless this process completes successfully.
Add	Select a process that will be performed on the pending value object created when the privilege that references this repository type is assigned to an entry. You can select a process of type Regular or Add Member Process from the current package or any other package in the same identity store, if the process is made public. Use the ... button to navigate to and add the process. This process will perform its operations, for example request an approval when assigning a role or create an authorization in a target application when assigning a privilege. Depending on the status of the operations, the role or privilege will or will not be assigned to the entry. The role or privilege is not assigned until the process has completed. Approvers for the Add Member Process are defined on the privilege.  Note If a Validate Add Process is defined, this Add Member Process will not be executed unless the Validate Add Process has completed successfully. Select or fill in a value in the Execute field to specify when the process should be executed relative to the valid from and valid to specified for the assignment. You have the following options: - When valid: Execute the process at the valid from specified in the assignment request. - Immediately: Execute the process when the assignment is requested. This is the default value for validate processes.

Field	Description
	Note This option works only for direct assignments of privileges. It does not work for privileges assigned through roles. Assignments of roles containing privileges are made as soon as the <i>valid from</i> date is reached, then the role and the privilege will be assigned. Number of days: In the <i>Execute</i> field, fill in a number of days before the <i>valid from</i> when the process should be executed. The <i>days before</i> description is displayed. For instance, if you enter <i>5</i>, the process will be executed 5 days before the date specified as <i>valid from</i> in the assignment request.
<i>Remove</i>	If the <i>Remove Member Process</i> is specified, the privilege that references this repository type will not be removed immediately, but a pending value object is created. You can select a process of type <i>Regular</i> or <i>Remove Member Process</i> from the current package or any other package in the same identity store, if the process is made public. Select the process to be executed on this object. Use the ... button to navigate to and add the process. The process will then for instance revoke the authorizations in the target system before a privilege is physically removed for the object. The role or privilege is not removed until the process has completed. Note If a <i>Validate Remove Process</i> is defined, this <i>Remove Member Process</i> will not be executed unless the <i>Validate Remove Process</i> has completed successfully.
<i>Modify Validity</i>	Select a process that will be performed on the pending value object created when the validity of an assignment is modified, either extended or shortened. You can select a process of type <i>Regular</i> or <i>Modify Member Process</i> from the current package or any other package in the same identity store, if the process is made public. Use the ... button to navigate to and add the process. This process will perform its operations, for example updating the validity in the target application. Depending on the status of the operations, the validity will or will not be modified. The validity is not modified until the process has completed. Note If a <i>Validate Modify Process</i> is defined, this <i>Modify Member Process</i> will not be executed unless the <i>Validate Modify Process</i> has completed successfully.
<i>Notification</i>	Select a process that will be used to send a notification that an assignment is about to expire. You can select a process of type <i>Regular</i> or <i>Notify Process</i> from the current package or any other package in the same identity store, if the process is made public. Use the ... button to navigate to and add the process.

Field	Description
	This process is normally specified to be executed a number of days before the assignment expires. This process may perform actions like: • Send an e-mail to the user that the assignment is about to expire • Extend the assignment by a given period of time • Create an approval to the manager Select or fill in a value in the <i>Execute</i> field to specify when the process should be executed relative to the <i>valid from</i> and <i>valid to</i> specified for the assignment. You have the following options: • When valid: Execute the process at the <i>valid from</i> specified in the assignment request. • Immediately: Execute the process when the assignment is requested. This is the default value for validate processes. Note This option works only for direct assignments of privileges. It does not work for privileges assigned through roles. Assignments of roles containing privileges are made as soon as the <i>valid from</i> date is reached, then the role and the privilege will be assigned. • Number of days: In the <i>Execute</i> field, fill in a number of days before the <i>valid from</i> when the process should be executed. The <i>days before</i> description is displayed. For instance, if you enter <i>5</i>, the process will be executed 5 days before the date specified as <i>valid from</i> in the assignment request.
<i>Provisioning</i>	Select the process that is to be executed when the privilege is assigned to an entry. You can select a process of type <i>Regular</i> or <i>Provisioning Process</i> from the current package or any other package in the same identity store, if the process is made public. Use the ... button to navigate to and add the process. If an <i>Add Member Process</i> is specified, the <i>Provisioning Process</i> will not be executed until the <i>Add Member Process</i> has completed. The process is executed on the user entry.
<i>Deprovisioning</i>	Select the process that is to be executed when the privilege is removed from an entry. You can select a process of type <i>Regular</i> or <i>Deprovisioning Process</i> from the current package or any other package in the same identity store, if the process is made public. Use the ... button to navigate to and add the process. This will normally be the opposite process to the <i>Provisioning Process</i>. If a <i>Remove Member Process</i> is specified, the <i>Deprovisioning Process</i> will not be executed until the <i>Remove Member Process</i> has completed. The process is executed on the user entry.
<i>Modify</i>	Select the process that is to be executed when any of the attributes of the connected entry (e.g. user) are modified. You can select a process of type <i>Privilege Modify Process</i>

Field	Description
	from a current package or any other package in the same identity store, if the process is made public. Use the ... button to navigate to and add the process. By default, the <i>Modify Process</i> is executed when any of the attributes are modified. The typical use case for this is to ensure that all user attributes are updated in the target repository(ies) when the user entry is modified.

Related Information

[Repository Type Properties \[page 1012\]](#)

[Configuring Event Processes for a Repository Type \[page 216\]](#)

[Member Event Handling \[page 81\]](#)

[Provisioning \[page 98\]](#)

9.1.9.5 Context (Repository Type)

Gives an overview of the properties on the *Context* tab of the repository type.

The *Context* tab of the repository type properties contains the following fields:

Field	Description
<i>Autoassign context policy</i>	The autoassign context policy decides how to handle default contexts defined for the users. You can choose between: • None: Do not use the default contexts defined for the user. • If missing: Only add default contexts if the assignment does not already have a context of the given type. • Always: Always add the automatically assigned context in addition to the any contexts already assigned.
<i>Context types</i>	The list contains all entry types that are defined as contexts. Select the check box in the <i>Allow</i> column to define allowed contexts for the repository type and the repositories based on it. Note If the list of context types is empty (i.e. no context types exist in the system), the check box in the <i>Allow</i> column will not be visible.

Related Information

[Repository Type Properties \[page 1012\]](#)

[Configuring Contexts for a Repository Type \[page 216\]](#)

[Context Based Assignments \[page 60\]](#)

9.1.10 Job Properties

Overview of the tabs of the job properties

The job properties contain the following tabs:

<i>General</i>	Enable or disable the job, rename or schedule a job.
<i>Logging</i>	Specify information about a job's logging. Note The <i>Logging</i> tab is also a part of the action task properties.
<i>State</i>	Shows information about the job's state. Note The <i>State</i> tab is also a part of the action task properties.
<i>Documentation</i>	Contains a text field only, for the comments and the documentation (description) of the job.

Related Information

[General \(Job\) \[page 1023\]](#)

[Logging \(Job/Action Task\) \[page 1024\]](#)

[State \(Job/Action Task\) \[page 1025\]](#)

[Documentation \(Job\) \[page 1027\]](#)

[Action Task \[page 1129\]](#)

9.1.10.1 General (Job)

Gives an overview of the properties on the [General](#) tab of the job.

The [General](#) tab of the job properties contains the following fields:

Field	Description
Enabled	Select this check box to enable or disable the job.
Job ID/Name	Displays the unique ID and the name of the job. You change name of the job in the tree view.
Schedule Rule	The default schedule rule when createing a job is On demand, which means that it must be started manually. Select a schedule rule that determines how often/when the job is scheduled to be run.
Schedule Time	The field displays the next time the job is scheduled to run. • Choose Run to schedule the job to be run immediately. The scheduled time is set to the current time. • Choose Stop to stop a running job.
Run by Dispatcher(s)	Select the dispatcher(s) that are allowed to run this job.
Bootstrap Job	There are two sets of job timeouts defined. The default job timeouts may be too short for the jobs that may take longer to process, for instance jobs that are used to bootstrap a system. Select this option to specify that this job is a bootstrap job that should use the bootstrap job timeouts.
Right Trim Source Attributes	Select this option to specify that you want to remove any trailing spaces from attribute values read from the source files/repositories of the passes in the job.

Related Information

[Selecting a Dispatcher for a Job \[page 296\]](#)

[Testing a Job \[page 299\]](#)

[Scheduling a Job \[page 295\]](#)

[Enabling or Disabling Jobs \[page 293\]](#)

[Job Properties \[page 1022\]](#)

9.1.10.2 Logging (Job/Action Task)

Gives an overview of the properties on the [Logging](#) tab of the job.

The [Logging](#) tab of the job properties contains the following fields:

Field	Description
Error Log File	The name of the error log file. This file will contain all messages and errors that occur during processing. The default value is <job name>.log . Choose the ... button to the right of the field if you want to browse the file system to find the directory where you want to create the file. If no file is specified, there will be no logging.
Reset Log File	Select this option to reset the log file each time the job is run. If you leave the check box empty, new entries will be appended to the existing log file.
Max No. of Errors to Tolerate	Specify the maximum number of errors to tolerate before the job is stopped. For example, if set to 1, and one error occurs, the job will continue to run. If a second error occurs, the job will stop. The default value is 0, which means no limit. If you don't want to tolerate any errors, you can write a script that will stop the job, and select this script in the Call Script in Case of Error field.
Max No. of Errors to Log in XML Log	The number of errors that should be written to the XML log.
Windows Event Log Level	Enter the level of error messages you want to include in the event log. You can choose from the following values: • No Logging No information is written to the event log. • Information/Warning/Error This is the maximum log level. Everything is written to the event log. This is the same information as is written to the Error log file.
SNMP Manager Address	Enter the numeric IP address to the SNMP manager that is to receive the SNMP messages from the job. To broadcast the messages, enter 255.255.255.255 as the address. Any SNMP manager within the local area network will receive the message.
SNMP Trap Level	Select the level of error messages you want to send as SNMP messages. You can choose from the following values: • No Logging No error messages are sent as SNMP messages. • Information/Warning/Error All messages are sent as SNMP messages. • Warning/Error Warnings and error messages are sent as SNMP messages.

Field	Description
	Error Only error messages are sent as SNMP messages.
Call Script in Case of Error	Enter the name of the script you want to call if an error occurs. You can write an error handler script, which will be called for every error. This script can include sending an e-mail to the system supervisor, for example. To edit the script, choose Edit... to the right of the field. If no script is specified, the New Script dialog box is displayed, and you can create a script.
Identity Management Log Level	Select log level for the Identity Management job log. You can choose between the following values: No Logging No information is written to the job log. Not even statistics. Information/Warning/Error All information is written to the job log. This is the default value.

Related Information

[Job Properties \[page 1022\]](#)

[Configuring a Job's Logging \[page 297\]](#)

[Viewing the Job Log \[page 300\]](#)

[Action Task \[page 1129\]](#)

[Configuring Logging for an Action Task \[page 249\]](#)

9.1.10.3 State (Job/Action Task)

Gives an overview of the properties on the [State](#) tab of the job.

The [State](#) tab of the job properties contains the following fields:

Field	Description
State	The job's state. It can be one of the following values: Idle The job is currently idle, waiting for the next scheduled run. Running The job is currently running. Stopping The job has been told to stop before it is completed.

Field	Description
	Error An error has occurred, and the job is aborted. The job can be rescheduled by choosing <i>Run now</i>. Disabled The job has been disabled, and will not be scheduled to run. Timeout The job has not been requested by any runtime engine within the specified amount of time (Idle timeout).
<i>Number of Executions</i>	This field shows how many times this job has been run.
<i>Job Last Started</i>	The field shows the date and time the job was last started.
<i>Job Last Finished</i>	The field shows the date and time the job was last finished.
<i>Scheduled Time</i>	 Note The field is not available for the action tasks. The field displays the next time the job is scheduled to run. - Choose <i>Run now</i> to schedule the job to be run immediately. The scheduled time is set to the current time. - Choose <i>Stop now</i> to stop a running job. - Choose <i>Force restart</i> to restart a stalled job.
<i>Status Code from Last Run</i>	The field shows the status code from the last run. <i>0</i> means <i>OK</i> .
<i>Status Text from Last Run</i>	The field shows the status text for the given status code from the last run of the job.
<i>Number of Errors</i>	This field displays the number of consecutive times the job failed during processing.
<i>Number of Start Attempts</i>	This field displays the number of consecutive times the job failed to start. This value is increased when the values for <i>Execution Timeout</i> or <i>Start Timeout</i> is exceeded.
<i>Entry Number</i>	The field displays the number of the entry being processed.
<i>Time</i>	The field shows the time that the runtime engine last reported that it was running.  Note The field is only filled in as long as the job is running.
<i>Total from Last Run</i>	The field displays the total number of entries processed by the last job.

Field	Description
<i>Dispatcher</i>	The field displays the name of the dispatcher that processes the job.
	 Note The field is only filled in as long as the job is running.

Related Information

[Viewing a Job's State \[page 307\]](#)

[Job Properties \[page 1022\]](#)

[Testing a Job \[page 299\]](#)

[Action Task \[page 1129\]](#)

[Viewing a State of an Action Task \[page 250\]](#)

9.1.10.4 Documentation (Job)

Use the *Documentation* tab in the job properties to document a job.

The *Documentation* tab of the job properties contains a text field where job documentation (e.g. job description and other comments) can be entered.

Related Information

[Job Properties \[page 1022\]](#)

[Entering a Description of a Job \[page 292\]](#)

9.1.11 Repository Job Properties

The repository job properties contain the following tabs:

<i>General</i>	Enable or disable the job, rename or schedule a job.
<i>Logging</i>	Specify information about a job's logging.

ⓘ Note

The [Logging](#) tab is also a part of the action task properties.

Documentation	Contains a text field only, for the comments and the documentation (description) of the job.
Advanced	Specify the advanced options for the runtime engine. Only visible if you choose the Advanced button from the General tab.

9.1.11.1 General (Repository Job)

The [General](#) tab of the job properties contains the following fields:

Field	Description
Enabled	Select this check box to enable or disable the job.
Job ID/Name	Displays the unique ID and the name of the job. You change name of the job in the tree view.
Repository Type	Displays the repository type that was selected when the job was created. ⓘ Note The repository type cannot be changed.
Run by Dispatcher(s)	Select the dispatcher(s) that are allowed to run this job.
Repositories	The list shows the repository of the selected repository type. You can select one of the repositories and choose one of the actions from the context menu. The table contains the following columns: • ID
Schedule Rule	Select a schedule rules for the job. New jobs will get the following default schedule rules: • Regular jobs: "On demand" • Provisioning jobs: "Provision"

Field	Description
	 Note When you transport a package, note that the schedule rule you have set for a repository job is not exported and imported with the package. In this case, the repository job is imported with its default schedule rule "On demand".
<i>Schedule Time</i>	The field displays the next time the job is scheduled to run. - Choose <i>Run now</i> to schedule the job to be run immediately. The scheduled time is set to the current time. - Choose <i>Stop now</i> to stop a running job.
<i>Bootstrap Job</i>	There are two sets of job timeouts defined. The default job timeouts may be too short for the jobs that may take longer to process, for instance jobs that are used to bootstrap a system. Select this option to specify that this job is a bootstrap job that should use the bootstrap job timeouts. Changing the <i>Bootstrap Job</i> option of the main repository job (either activate or deactivate) does not affect jobs already run for specific repositories. For example, if you have run jobs with repository A and repository B and then you activate the option, bootstrap timeouts will not be applied to those jobs. The alternative is to copy the main repository job, activate the <i>Bootstrap Job</i> option and run it for one specific repository.
<i>Right Trim Source Attributes</i>	Select this option to specify that you want to remove any trailing spaces from attribute values read from the source files/repositories of the passes in the job.

9.1.12 Pass Properties

A job consists of one or more passes that are responsible for performing a specific action, like reading from or writing to a data source (repository) or some other task.

There are two types of passes:

- From-passes
- To-passes

Related Information

[Jobs and Passes \[page 78\]](#)

[Maintaining Passes \[page 307\]](#)

[Adding a Pass to a Job \[page 308\]](#)

[Viewing and Configuring Pass Properties \[page 308\]](#)

9.1.12.1 Common Tabs

Some tabs in the pass properties are common for all passes (i.e. both From- and To-passes).

The common tabs are:

- [General](#) tab
- [Documentation](#) tab

9.1.12.1.1 General (Passes)

Gives an overview of the properties on the [General](#) tab of the pass properties.

The [General](#) tab of the pass properties contains the following fields:

Pass type	The field displays the type of the pass.
Repository	Select the repository for the pass. The value --None/Inherited-- is defined by default. This repository is used as follows: • If the pass belongs to a regular job, the repository selected for the pass is used. • If the pass belongs to a repository job, the repository selected for the repository job is used. The repository selected for the pass is not taken into account. • If the pass belongs to an action task that is part of a process, this field is disabled, and it is not possible to select a repository for the pass. The repository selected for the process is used.
Initialization script	Initialization scripts are used to perform special actions before the pass is executed, for instances to prepare the source of the pass. The script has no parameters or return values. Note This field is not available for the From Generic pass.
Termination script	Termination scripts are executed after the pass is run. The script has no parameters or return values.

ⓘ Note

This field is not available for the From Generic pass.

Entry script

Entry scripts are used for pre-processing of each entry before they are written to the target.

ⓘ Note

This field is not available for the From Generic pass.

Related Information

[Defining a Repository for a Pass \[page 313\]](#)

[Defining Custom Scripts for a Pass \[page 314\]](#)

[Viewing and Configuring Pass Properties \[page 308\]](#)

9.1.12.1.2 Documentation (Passes)

Use the [Documentation](#) tab in the pass properties to document a pass.

The [Documentation](#) tab contains a text field where the pass documentation (e.g. pass description and other comments) can be entered.

Related Information

[Entering a Description of a Pass \[page 380\]](#)

[Pass Properties \[page 1029\]](#)

9.1.12.2 From-Passes

From-passes are used to read from a data source and write to the temporary database.

There are From-passes that are used to read from different types of data sources:

- From ASCII file
- From Database
- From LDAP directory
- From Custom

- From LDIF file
- From Generic

Common Tabs

Two of the tabs in From-pass properties are the same (common) for all From-passes:

- [Destination](#) tab
- [Delta](#) tab

9.1.12.2.1 From ASCII File

The From ASCII file pass is used to read from ASCII files in the file system.

The pass reads the ASCII file in the following format:

- One record per line.
- Common separator between the fields in the record.
- The separator between multiple values must be different from the field separator.
- The first line in the ASCII file must contain the names of the columns.

One variant of ASCII files is CSV files (Comma Separated Values), although the runtime engine does not require that the field separator is a comma. Most applications are able to export their data into CSV format.

The properties of a From ASCII file pass contain the following tabs:

General	Select repository for the pass or define scripts. General (Passes) [page 1030]
Source	Define the source file. Source (From ASCII File Pass) [page 1036]
Destination	Define the information about the temporary database and the pass definitions. Destination (From-Passes) [page 1046]
Delta	Enable and configure delta. Delta (From-Passes) [page 1047]
Documentation	Contains a text field only, for the comments and the documentation (description) of the pass. Documentation (Passes) [page 1031]

Related Information

[Specifying the Source for a From ASCII File Pass \[page 315\]](#)

9.1.12.2.2 From Database

The From Database pass is used when reading information from a database or other data source.

The Java runtime engine is able to read any data source referenced by a JDBC URL (JDBC connection). The properties of a From Database pass contain the following tabs:

<i>General</i>	Select repository for the pass or define scripts. General (Passes) [page 1030]
<i>Source</i>	Define the source database. Source (From Database Pass) [page 1038]
<i>Destination</i>	Define the information about the temporary database and the pass definitions. Destination (From-Passes) [page 1046]
<i>Delta</i>	Enable and configure delta. Delta (From-Passes) [page 1047]
<i>Documentation</i>	Contains a text field only, for the comments and the documentation (description) of the pass. Documentation (Passes) [page 1031]

Related Information

[Specifying the Source for a From Database Pass \[page 316\]](#)

9.1.12.2.3 From Custom

Custom passes provide an open interface for extending the Identity Center to access data sources that are not accessible from other passes.

Custom passes differ from Generic passes (To Generic and From Generic) in that they are developed in Java, making them more efficient. They can use interfaces that are not accessible from scripts.

The *Pass type* of a custom class (field in the *Source* tab) is the base name of the class file containing the custom pass. Custom passes are configured using a parameter table (in the *Source* tab) consisting of

parameter name - parameter value pairs. The parameter names differ from pass to pass, and you will need to obtain configuration details from the developer of the Java class.

Note

For details about this pass type, contact SAP. There are Java classes that can be used to read data from an identity store, a JMS queue, JNDI and XML files.

The properties of a From Custom pass contain the following tabs:

<i>General</i>	Select repository for the pass or define scripts. General (Passes) [page 1030]
<i>Source</i>	Define the source. Source (From Custom Pass) [page 1039]
<i>Destination</i>	Define the information about the temporary database and the pass definitions. Destination (From-Passes) [page 1046]
<i>Delta</i>	Enable and configure delta. Delta (From-Passes) [page 1047]
<i>Documentation</i>	Contains a text field only, for the comments and the documentation (description) of the pass. Documentation (Passes) [page 1031]

Related Information

[Specifying the Source for a From Custom Pass \[page 317\]](#)

[From Generic \[page 1035\]](#)

[To Generic \[page 1053\]](#)

9.1.12.2.4 From LDAP Directory

A From LDAP directory pass is used to read information from a directory server, using the LDAP protocol.

The properties of a From LDAP directory pass contain the following tabs:

<i>General</i>	Select repository for the pass or define scripts. General (Passes) [page 1030]
<i>Source</i>	Define the source LDAP directory server.

	Source (From LDAP Directory Pass) [page 1041]
<i>Destination</i>	Define the information about the temporary database and the pass definitions. Destination (From-Passes) [page 1046]
<i>Delta</i>	Enable and configure delta. Delta (From-Passes) [page 1047]
<i>Documentation</i>	Contains a text field only, for the comments and the documentation (description) of the pass. Documentation (Passes) [page 1031]

Related Information

[Specifying the Source for a From LDAP Directory Pass \[page 318\]](#)

9.1.12.2.5 From Generic

This pass type is used to read data sources not supported by any other pass type, for instance Lotus Notes.

The properties of a From Generic pass contain the following tabs:

<i>General</i>	Select repository for the pass or define scripts. General (Passes) [page 1030]
<i>Source</i>	Define the source. Source (From Generic Pass) [page 1045]
<i>Destination</i>	Define the information about the temporary database and the pass definitions. Destination (From-Passes) [page 1046]
<i>Delta</i>	Enable and configure delta. Delta (From-Passes) [page 1047]
<i>Documentation</i>	Contains a text field only, for the comments and the documentation (description) of the pass. Documentation (Passes) [page 1031]

Related Information

[Specifying the Source for a From Generic Pass \[page 321\]](#)

9.1.12.2.6 From LDIF File

From LDIF file passes are used to read data from an LDIF file produced by either the Identity Center or other tool.

The properties of a From LDIF file pass contain the following tabs:

<i>General</i>	Select repository for the pass or define scripts. General (Passes) [page 1030]
<i>Source</i>	Define the source file. Source (From LDIF File Pass) [page 1046]
<i>Destination</i>	Define the information about the temporary database and the pass definitions. Destination (From-Passes) [page 1046]
<i>Delta</i>	Enable and configure delta. Delta (From-Passes) [page 1047]
<i>Documentation</i>	Contains a text field only, for the comments and the documentation (description) of the pass. Documentation (Passes) [page 1031]

Related Information

[Specifying the Source for a From LDIF File Pass \[page 322\]](#)

9.1.12.2.7 Source (From ASCII File Pass)

Gives an overview of the properties on the [Source](#) tab of the From ASCII File pass properties.

The [Source](#) tab of the pass properties contains the following fields:

<i>File name</i>	Specify the file name of the source file. Choose the ... button to the right of the field if you want to browse the file system

	to find the correct file and directory. You can also specify remote files using the http and ftp protocols (http:// and ftp://).
<i>Field separator</i>	Specify the character that separates fields in the source file. If you want to use a space as the delimiter, use the word SPACE (without quotes). If you want to use a tab character as the delimiter, use the word TAB.
<i>Fixed record length</i>	Select this option to specify that each column in the file has a fixed record length. You must specify record lengths in the <i>Definitions table</i> in the <i>Destination</i> tab.
<i>Header line</i>	Select this option to specify whether the source file has a header line or not. Leave the check box empty if you process a CSV file or a source file with fixed record length. The sequence of the fields in the <i>Definitions table</i> is used to process the source file.
<i>Character set</i>	Select a character set in the list to specify the character set of the file to be read: • <i>Auto</i> The runtime engine determines the character set based on the default character set specified in the Java environment. The runtime engine also attempts to recognize the character set used in the file (by inspecting the contents). If the file contains BOM (byte order mark) tags, it will be processed correctly. For best results, specify the character set to be used. • <i>ANSI (ISO-8859-1)</i> • <i>UTF-8</i> Note If you are writing the information to a database, also make sure to use a data type for the database that supports the character set of the source file, for instance VARCHAR vs NVARCHAR. For more information about data types and character encoding, see the documentation for your database system.

Related Information

[Specifying the Source for a From ASCII File Pass \[page 315\]](#)

9.1.12.2.8 Source (From Database Pass)

Gives an overview of the properties on the [Source](#) tab of the From Database pass properties.

The [Source](#) tab of the pass properties contains the following fields:

Database	Specify the data source to be processed (enter an JDBC URL). You can use the JDBC connection wizard to create the URL. Choose ... to use the wizard and fill in the necessary parameters.
Encrypt Connection String	Select this option to specify that the connection string should be stored encrypted in the job configuration file. It will be displayed using asterisks in the field.
SQL Query	This is the SQL query used to read the information. Only SELECT statements may be used. Depending on the data source, you may need to prepare it in different ways. For instance, to read data from Excel, you need to name an area in the spreadsheet to be able to refer to it. Note The SQL query should not be terminated or delimited with semicolon. Note When using Microsoft SQL Server, it is recommended that all SELECT queries should include "with (nolock)". This will reduce the table locking, improve performance, and reduce the probability of deadlocks. For instance: <pre>SELECT <columns> FROM <table> WITH (NOLOCK) WHERE <column>=<value></pre> If you specified an XML recordset file in the Database field, you only specify the WHERE clause, for instance: <pre>(cn like 'a%')</pre> This will then be used as a filter for processing the XML recordset file. To insert constants, scripts and system parameters, right-click and select the available options from the context menu.
Multivalue Separator	Enter the character that separates values in fields containing multiple values in the source file. If the separator is space or tab, enter the words SPACE or TAB (without quotes).

Related Information

[Specifying the Source for a From Database Pass \[page 316\]](#)

9.1.12.2.9 Source (From Custom Pass)

Gives an overview of the properties on the [Source](#) tab of the From Custom pass properties.

When configuring a From Custom pass, you use the [Source](#) tab to enter the necessary parameters for your Java class. The [Source](#) tab of the pass properties contains the following fields:

Pass type	Enter the name of the Java class to be called. The class must be accessible with the given CLASSPATH.
Parameter table	Enter the parameter names and the corresponding values in the table. These parameters are submitted to the class when it is called.

 **Note**
The parameter names are case-sensitive.

Related Information

[Specifying the Source for a From Custom Pass \[page 317\]](#)

9.1.12.2.9.1 Parameters for FromSFSFSoap Pass

Gives an overview of the FromSFSFSoap pass specific parameters.

When configuring a FromSFSFSoap pass, you use the [Source](#) tab to enter the parameter names and the corresponding values in the parameter table. These parameters are submitted to the class when it is called.

Most of the parameters values are repository constants that are taken from the repository, using the following syntax:

```
%%$rep.<constant>%
```

Parameter	Value
USERNAME	%%\$rep.SFSF_USER%
PASSWORD	%%\$rep.SFSF_PASSWORD%
SFSF_PROTOCOL	%%\$rep.SFSF_PROTOCOL%
SFSF_HOST	%%\$rep.SFSF_HOST%
SFSF_PORT	%%\$rep.SFSF_PORT%

Parameter	Value
PROXY_HOST	rep.PROXY_HOST
PROXY_PORT	rep.PROXY_PORT
SFSF_QUERY	This parameter allows you to specify the information that will be returned by the API. Possible values are listed in the official documentation. See For example: Implementing the Employee Central Compound Employee API person, personal_information, address_information, email_information, employment_information, job_information, phone_information
SFSF_SOAP_USED_API	CompoundEmployee
TRUSTSTORE	rep.TRUSTSTORE
TRUSTSTORE_PASSWORD	rep.TRUSTSTORE_PASSWORD
CONNECT_TIMEOUT	rep.CONNECT_TIMEOUT
READ_TIMEOUT	rep.READ_TIMEOUT
PROXY_USER	rep.PROXY_USER
PROXY_PASSWORD	rep.PROXY_PASSWORD
SFSF_SOAP_FILTER	This parameter allows you to specify filter(s) to the query. ❖ Example Specifying one filter to the query Company= 'SAP' ❖ Example Specifying multiple filters to the query isContingentWorker IN ('true','false') AND Company= 'SAP'

Parameter	Value
SFSF_SOAP_PARAMETERS	This parameter allows you to specify additional parameters to the SOAP query. If you specify more than one parameter, use double pipe () as delimiter. For example: param1=paramValue1 param2= paramValue2 You can use this parameter, for example, to read multiple events that occur on the same date for a specific employee. In this case, set the value to resultOptions=allJobChangesPerDay.

Related Information

[Success Factors Connector Package \[page 457\]](#)

9.1.12.2.10 Source (From LDAP Directory Pass)

Gives an overview of the properties on the [Source](#) tab of the From LDAP directory pass properties.

Note

LDAP version 2 is the protocol for communication with the directory server. When using this protocol, only entries in Latin characters are read correctly. Make sure you do not use non-Latin characters as they are read incorrectly.

The [Source](#) tab of the pass properties contains the following fields:

LDAP URL

Enter the URL containing the operation to be performed in the directory. The URL also contains the address of the server to be accessed.

Note

To make it possible to read a branch in the directory, you must include the search type and filter in the URL.

Choose the ... button to the right of the field to open a dialog box where you can specify the URL.

For detailed information about LDAP URLs, see RFC1959.

Note

If you use an SQL query to retrieve the search filters from the database, use the variable as part of the filter and not starting point in the LDAP URL to avoid opening a new connection to the directory server for each value returned from the database.

Directory login name Enter the DN used to log on to the directory server. If you leave the field empty, anonymous login is used.

Note

If you select *Kerberos* as the security option in the *Security option* field and the fields *Directory login name* and *Directory login password* are empty, the user will be authenticated using the current login credentials. This only applies to Active Directory or other directory servers supporting the Kerberos protocol.

Directory login password Enter the login password. It will not be displayed on the screen.

Note

This parameter is stored as unencrypted in the .DSE file, and will be visible to anyone with the access to this file. Make sure that the file is access protected.

Directory size limit Specify the LDAP size limit when performing search operations.

Note

If the server has a size limit, and this is lower than the client's limit, the server size limit will be used.

Directory time limit Specify the LDAP time limit for performing search operations.

Note

If the server has a time limit, and this is lower than the client's limit, the server time limit will be used.

Directory page size LDAP servers may be able to return a specified number of entries in one search, and then continue from this point to return the next batch of entries. This feature is called paging.

Enter the number of entries you want the LDAP server to return in one page. The process continues until the size or time limit is reached. The behavior of the LDAP server may vary according to the server software.

If you enter 0 or leave the field empty, paging is not used.

Advanced Choose this button to open the *Advanced* tab where you can specify advanced parameters for resolving time and size limits if the directory server does not support paging.

The Identity Management can try to resolve the problems that arise if a pass attempts to retrieve more entries than the maximum allowed by the size or time limits set by the server.

Note

To be able to use size/time limit resolving, the LDAP URL has to include one and only one asterisk (*), for instance (sn=*).

- [Try to Resolve Size/Time Limit Problems](#)
Select this checkbox to enable the function.
- [Characters to Process in Case of Size/Time Limit](#)
In this field you specify how the runtime engine should break up the character sequence in order to resolve size/time limit problems.
- [Maximum Number of Recursive Loops](#)
Fill in this field with a number that indicates how many times the filter should be tried. This parameter is used to prevent infinite loops. In practice, the number 5 should be sufficient for most data sources, unless the data volume is very large and a great deal of data has the same 5 first characters as the search parameter.
- [From Database](#)
Enter the name of the database where the SQL queries are stored. When using the Java runtime engine the database must be referenced using an JDBC URL. You can use the JDBC URL wizard to create the URL.
You may also encrypt the provided connection string by selecting the option [Encrypt Connection String](#).
- [From SQL Statement](#)
Enter the SQL statement you want to use to retrieve the search filter from the database. If the pass reads from the identity store, it is recommended to use one of the views installed with the product. The column names in the database query will replace any matching variables (%<ColumnName>%) in the LDAP URL.
For instance, if you want to replace the host part of the URL with the result of an SQL query, you can enter an SQL query: **SELECT host FROM hosts**

Note

When using Microsoft SQL Server, it is recommended that all SELECT queries should include "with (nolock)". This will reduce the table locking, improve performance, and reduce the probability of deadlocks. For instance:

```
SELECT <columns> FROM <view> WITH (NOLOCK) WHERE  
<column>= '<value>'
```

The LDAP URL may look like this: **LDAP://%host%/o=myorg??SUB?(sn=*)**

Or if you have a table of e-mail addresses and want to retrieve the corresponding LDAP entries, you can enter an SQL query such as: **SELECT mailaddr FROM mymailinglist**

The LDAP URL may look like this: **LDAP://host.domain.com/o=myorg??SUB?(mail=%mailaddr%)**

Note

To avoid opening a new connection to the directory server for each value returned from the database, use the variable as part of the filter and not starting point in the LDAP URL.

<i>Character set</i>	A character set that the LDAP server uses.
<i>Follow referrals</i>	Use this option to specify how referrals are to be handled: • When the option is selected, the runtime engine returns the referred entry for all referrals that are encountered. • When the option is deselected, the runtime engine returns an error message for all referrals that are encountered.
<i>Security option</i>	Use this list to specify the security level and protocol for communication with the directory server. You can select one of the following values: • <i>Simple authentication</i> The login name and login password are sent unencrypted to the directory server. This is the default mode of operation. • <i>SSL</i> If the directory server you are connecting to supports SSL (Secure Sockets Layer) communication, you can specify that the runtime engine should use this connection when reading or writing data. The default port number when using SSL is 636. Ensure that you specify the correct port number in the URL in the <i>LDAP URL</i> field. Using SSL encrypts the communication channel to protect sensitive data. After the secure communication channel is set up, the login name and password are submitted to the server. • <i>Kerberos</i> This option is only relevant to communication with Active Directory servers or other directory servers supporting the Kerberos protocol. The current user's user name and password are then used to log on to the directory server. In this case, it is not necessary to fill in the fields <i>Directory login name</i> and <i>Directory login password</i>.
<i>Alias resolving</i>	Some directory servers contain aliases. Aliases are directory objects that refer to a "normal" object, for instance a person. The value in the list is submitted to the directory server, and the behavior depends on the directory server's interpretation of the parameter.

Note

An alias can only be resolved if it is found on the same server, using the starting point configured for that server.

The values have the following meaning:

- *Never*
Alias nodes are never resolved. The alias properties are used.
- *Searching*
Alias nodes are resolved in search operations.

Note

This value is not recommended, because under normal circumstances, this would be the same as selecting Always, but may produce unpredictable results.

- *Finding*
Aliases are resolved on BASE searches.
- *Always*
Aliases are always resolved.

Related Information

[Specifying the Source for a From LDAP Directory Pass \[page 318\]](#)

[Using Advanced Search Criteria \(From LDAP Directory Pass\) \[page 318\]](#)

9.1.12.2.11 Source (From Generic Pass)

This pass type is used to read data from data sources not supported by any other pass type, for instance Lotus Notes.

The *Source* tab of the pass properties contains the following fields:

<i>Open data source</i>	The first part of this field is the name of the script you want to run to open the data source. The second field contains any input parameter to the script. The script must return nothing if the processing was successful. If an error occurs during execution, the script must return a string containing the error message to be added to the log file, and the pass is aborted.
<i>Next data entry</i>	The script is used to read the entries in the data source. The first part of this field is the name of the script. The second field contains any input parameter to the script. The script is run once for each entry in the data source. The script must return a dictionary object containing the name and value of each attribute for the processed entry. The script will be called again in a loop as long as it returns a dictionary object. When all entries are processed, the loop is stopped by returning nothing from the script. If the script fails, it should return nothing, and the pass is aborted.
<i>Close data source</i>	The script is used to close the data source. The first part of this field is the name of the script. The second field contains any input parameters to the script.

The script must return nothing if the processing was successful. If an error occurs during execution, the script must return a string containing the error message to be added to the log file, and the pass is aborted.

Related Information

[Specifying the Source for a From Generic Pass \[page 321\]](#)

9.1.12.2.12 Source (From LDIF File Pass)

Gives an overview of the properties on the *Source* tab of the From LDIF file pass properties.

When configuring a From LDIF file pass, you use the *Source* tab to enter the information about the source of the pass. The tab contains the following fields:

<i>File name</i>	File name of the source file.
<i>LDIF character set</i>	Character set of the LDIF file. You can choose between the following values: • <i>ANSI</i> • <i>UTF-8</i> • <i>ANSI DN/UTF-8 attr</i>

Related Information

[Specifying the Source for a From LDIF File Pass \[page 322\]](#)

9.1.12.2.13 Destination (From-Passes)

Gives an overview of the properties on the *Destination* tab of the From-passes.

Use the *Destination* tab to enter information about the destination of the pass, the temporary database. The *Destination* tab of the pass properties contains the following fields:

<i>Identity Center database</i>	Select this option to use the Identity Center database as the temporary destination database. This is set as a default for the pass.

Database	Select this option to use other database (not the Identity Center database). Enter the JDBC connection for the database. Choose ... to use the JDBC connection wizard and fill in the necessary parameters.
Encrypt connection string	Select this option to specify that the connection string should be stored encrypted in the job configuration file. It will be displayed using asterisks in the field. This option is only enabled when the Database check box is selected.
Table name	Specify the name of the database table in the temporary database that will contain the data.
Table update	The value in this list determines whether or not the table in the temporary database is deleted before the data is loaded.
Definitions table	Use the definitions table to enter the definitions for the pass. You can use a template to simplify the work with the pass definitions (Insert template button).

Related Information

[Specifying the Temporary Database \(Destination\) for From-Passes \[page 323\]](#)

[Specifying Pass Definitions for From-Passes \[page 324\]](#)

9.1.12.2.14 Delta (From-Passes)

Use the [Delta](#) tab to enter information about the delta for the passes.

The [Delta](#) tab in the properties of the From-passes contains the following fields:

Enable delta	Select this option to specify that you want to create a delta for this pass. If this option is selected, the fields SAP Identity Management Database or Database , and Delta identifier must be filled in.
SAP Identity Management database	Select this option to use the SAP Identity Management database as the delta database. If this option is selected, the fields Database and Encrypt connection string are disabled.

<i>Database</i>	Select this option to use other database (not the Identity Management database) as the delta database. Enter the JDBC connection for the database. Choose ... to use the JDBC connection wizard and fill in the necessary parameters.
<i>Encrypt connection string</i>	Select this option to specify that the connection string should be stored encrypted in the job configuration file. It will be displayed using asterisks in the field. This option is only enabled when the <i>Database</i> check box is selected.
<i>Delta identifier</i>	The delta database may contain delta information from more than one pass. This identifier is added to make the delta of each pass unique.
<i>Delta key</i>	The field contains the entries' key in the delta database. The field is used in two ways, depending on the pass type: - For To ASCII file passes, Shell execute passes and To Database passes where the option <i>SQL updating</i> in the <i>Destination</i> tab is selected, you enter the delta key here. Use the same syntax as when defining the delta key for other pass types. - For all other pass types, the delta key is specified as the first line of the pass definition (the naming attribute). The delta key is displayed here, but cannot be modified.
<i>Source key</i>	This is a unique reference to the entry in the data source. Enter the column in the source that contains the unique key as a recordset variable. You can combine several columns to create a unique key. You can also enter any text or reference any variable or constant. If the data source is a database that the runtime engine is able to access, the context menu contains all columns in the data source.
<i>Generate delta only</i>	Select this option to produce the delta file only. No data is updated in the destination. This function can, for instance, be used for testing purposes, to regenerate a delta file or to produce a delta database for statistical purposes.
<i>Skip unchanged entries</i>	This option indicates whether to check for unchanged entries or not. If this option is not selected, all entries will be written to the target every time the job is executed. If it is selected, only the modified or new entries will be updated with the log information, although the delta will not be used when updating the target repository. This option is selected by default. Note When using a delta database in From-passes, selecting this option implies that unmodified entries in the data source will not be added to the temporary database. If this option is not selected, all entries are written to the temporary database, and the delta database is updated with the log information.
<i>Mark for deletion</i>	Select this option to specify that the runtime engine should mark entries as deleted. In the delta database, the operation code, fingerprint, etc. of each entry is updated when the job is run. New entries are added. To identify deleted entries, the delta database is searched to find missing entries and their operation code is set accordingly.

As this operation can be time-consuming, you can turn it off if it is not necessary to identify deleted entries, for instance in a To ASCII file pass.

<i>Max limit for mark for deletion</i>	Enter the maximum percentage of entries that may be marked for deletion. If this limit is exceeded, nothing is marked for deletion in the delta database. This is to prevent deletions caused by errors in the input data.

<i>Max real updates</i>	Enter the number of entries that can be modified with this pass. If the limit is reached, the pass stops and the job continues with the other passes. This can be used to spread the load when performing large updates, for instance during three successive nights.

Note

Make sure this function is used together with a delta database that takes care of modification and that the data is not too dynamic in relation to the number of updates you specify.

<i>Audit trail level</i>	You can choose from three levels: • <i>None</i> No audit trail is made. • <i>Entry level</i> Information about each entry modified is stored in the delta database. • <i>Attribute level</i> Information about each modified attribute is logged.

Note

A detailed audit trail level may reduce performance and increase the size of the delta database.

<i>Max number of entries in Audit trail table</i>	You can control the size of the audit trail table by defining a maximum number of entries.

Note

If you leave the field empty or enter 0, the table size may increase dramatically.

Related Information

[Configuring Delta for Passes \[page 379\]](#)

[The Delta Database \[page 1078\]](#)

9.1.12.2.15 Prefix Options (From-Passes)

In the specification of the definitions for a From-pass, the first column in the definitions table can contain a prefix that indicates special treatment of the following line.

The following prefix options are available:

<i>Indexed column (+)</i>	The current column will be indexed and will allow duplicates. This prefix is often used to optimize preceding Select statements.
<i>Indexed column - no duplicates (*)</i>	The same as the <i>Indexed column (+)</i> prefix, but does not allow duplicates.
<i>Except from delta database (@)</i>	Indicates that the column is excluded from the fingerprint generation for the delta database.
<i>Disable (#)</i>	Indicates that the line is a comment.

Related Information

[Specifying Pass Definitions for From-Passes \[page 324\]](#)

9.1.12.2.16 The Delta Key (From-Passes)

The first (blue) line of the pass definition (the naming attribute) is used to define the key in the delta database.

The delta key is displayed read-only in the *Delta key* field on the *Delta* tab.

Any unique attribute can be used as the delta key, such as the e-mail address or employee number. Note that names are not normally unique.

Related Information

[The Naming Attribute \[page 1083\]](#)

[The Delta Key \(To-Passes\) \[page 1073\]](#)

9.1.12.3 To-Passes

To-passes are used to read the data from the temporary database and into the target system.

There are To-passes that are used to read data into different types of target systems:

- To Identity store
- To ASCII file
- To Database
- To Generic
- To LDAP directory
- To LDIF or DSML file
- To Custom
- Shell execute

Common Tabs

Two of the tabs in To-pass properties are the same (common) for all To-passes:

- [Source](#) tab
- [Delta](#) tab

9.1.12.3.1 To Identity Store

This pass type is used to update an identity store.

The properties of a To Identity store pass contain the following tabs:

General	Select repository for the pass or define scripts. General (Passes) [page 1030]
Source	Define the source. Source (To-Passes) [page 1057]
Destination	Define the information about the destination for the pass. Destination (To Identity Store Pass) [page 1059]
Delta	Enable and configure delta. Delta (To-Passes) [page 1070]
Documentation	Contains a text field only, for the comments and the documentation (description) of the pass.

Note

The validation of the provided attribute values against the allowed values is done only on the defined in the legal values *Value list*. For more information, see [Attribute Values \(Attribute\) \[page 999\]](#).

9.1.12.3.2 To ASCII File

This pass is used to write data to an ASCII output file. You can use it to generate reports, batch files, CSV files, HTML files, etc.

Use two To ASCII file passes to make a header row in the ASCII file you produce. Select option *Reset output file* (in the *Destination* tab) only for the first pass.

The properties of a To ASCII file pass contain the following tabs:

<i>General</i>	Select repository for the pass or define scripts. General (Passes) [page 1030]
<i>Source</i>	Define the source. Source (To-Passes) [page 1057]
<i>Destination</i>	Define the information about the destination for the pass. Destination (To ASCII File Pass) [page 1061]
<i>Delta</i>	Enable and configure delta. Delta (To-Passes) [page 1070]
<i>Documentation</i>	Contains a text field only, for the comments and the documentation (description) of the pass. Documentation (Passes) [page 1031]

9.1.12.3.3 To Database

This pass is used to update a database or any other data source that can be accessed using a JDBC URL. Normally, the data source needs to be defined before you can update it.

The properties of a To Database pass contain the following tabs:

<i>General</i>	Select repository for the pass or define scripts.

	General (Passes) [page 1030]
<i>Source</i>	Define the source. Source (To-Passes) [page 1057]
<i>Destination</i>	Define the information about the destination for the pass. Destination (To Database Pass) [page 1063]
<i>Delta</i>	Enable and configure delta. Delta (To-Passes) [page 1070]
<i>Documentation</i>	Contains a text field only, for the comments and the documentation (description) of the pass. Documentation (Passes) [page 1031]

9.1.12.3.4 To Generic

This pass is used to update data repositories not supported by any other pass type.

The properties of a To Generic pass contain the following tabs:

<i>General</i>	Select repository for the pass or define scripts. General (Passes) [page 1030]
<i>Source</i>	Define the source. Source (To-Passes) [page 1057]
<i>Destination</i>	Define the information about the destination for the pass. Destination (To Generic Pass) [page 1064]
<i>Delta</i>	Enable and configure delta. Delta (To-Passes) [page 1070]
<i>Documentation</i>	Contains a text field only, for the comments and the documentation (description) of the pass. Documentation (Passes) [page 1031]

9.1.12.3.5 To LDAP Directory

This pass is used to update a directory server.

During the development phase, it is often a good idea to generate the LDIF output, so you can see the result of the pass. When the job works correctly you can simply change it to a To LDAP directory pass.

The properties of a To LDAP directory pass contain the following tabs:

<i>General</i>	Select repository for the pass or define scripts. General (Passes) [page 1030]
<i>Source</i>	Define the source. Source (To-Passes) [page 1057]
<i>Destination</i>	Define the information about the destination for the pass. Destination (To LDAP Directory Pass) [page 1065]
<i>Delta</i>	Enable and configure delta. Delta (To-Passes) [page 1070]
<i>Documentation</i>	Contains a text field only, for the comments and the documentation (description) of the pass. Documentation (Passes) [page 1031]

9.1.12.3.6 To LDIF or DSML File

This pass is used to write data to the LDIF output file.

LDIF is defined in RFC2849 *The LDAP Data Interchange Format (LDIF) - Technical Specification*. You can also create a file in DSML (Directory Services Markup Language) format using this pass type. (For more information about DSML, see <http://www.dsml.org>)

Note

A To LDAP directory pass is recommended rather than a To LDIF or DSML file. However, during the development phase, it is often a good idea to generate the LDIF output, so you can see the result of the pass. When the job works correctly you can simply change it to a To LDAP directory pass.

The properties of a To LDIF or DSML file pass contain the following tabs:

<i>General</i>	Select repository for the pass or define scripts. General (Passes) [page 1030]
<i>Source</i>	Define the source.

	Source (To-Passes) [page 1057]
<i>Destination</i>	Define the information about the destination for the pass. Destination (To LDIF or DSML File Pass) [page 1067]
<i>Delta</i>	Enable and configure delta. Delta (To-Passes) [page 1070]
<i>Documentation</i>	Contains a text field only, for the comments and the documentation (description) of the pass. Documentation (Passes) [page 1031]

9.1.12.3.7 To Custom

Custom passes provide an open interface for extending the Identity Center to access data sources not accessible from other passes.

Custom passes differ from Generic passes (To Generic and From Generic) in that they are developed in Java, making them more efficient. They can use interfaces not accessible from scripts.

The pass type of a custom class (the field *Pass type* in the *Destination* tab) is the base name of the class file containing the custom pass. Custom passes are configured using a table of parameter name - parameter value pairs. The parameter names differ from pass to pass, and you will have to obtain configuration details from the developer of the Java class.

Note

For details about this pass type, contact SAP. There are Java classes that can be used to write to an identity store, a JMS queue and to JNDI.

The properties of a To Custom pass contain the following tabs:

<i>General</i>	Select repository for the pass or define scripts. General (Passes) [page 1030]
<i>Source</i>	Define the source. Source (To-Passes) [page 1057]
<i>Destination</i>	Define the information about the destination for the pass. Destination (To Custom Pass) [page 1068]
<i>Delta</i>	Enable and configure delta. Delta (To-Passes) [page 1070]

[Documentation](#)

Contains a text field only, for the comments and the documentation (description) of the pass.

[Documentation \(Passes\) \[page 1031\]](#)

Related Information

[From Generic \[page 1035\]](#)

[To Generic \[page 1053\]](#)

9.1.12.3.8 Shell Execute

This pass is used to run a file, possibly with parameters from the job.

This means that the Identity Center may start other applications and batch jobs, which perform external operations. Examples of such operations include:

- Sending data using e-mail
- Printing information
- Running an application to extract information from a data source

The properties of a Shell execute pass contain the following tabs:

[General](#)

Select repository for the pass or define scripts.

[General \(Passes\) \[page 1030\]](#)

[Source](#)

Define the source.

[Source \(To-Passes\) \[page 1057\]](#)

[Destination](#)

Define the information about the destination for the pass.

[Destination \(Shell Execute Pass\) \[page 1069\]](#)

[Delta](#)

Enable and configure delta.

[Delta \(To-Passes\) \[page 1070\]](#)

[Documentation](#)

Contains a text field only, for the comments and the documentation (description) of the pass.

[Documentation \(Passes\) \[page 1031\]](#)

9.1.12.3.9 Source (To-Passes)

Gives an overview of the properties on the [Source](#) tab of the To-passes.

Use the [Source](#) tab to enter information about the source of the pass, the temporary database. The [Source](#) tab of the pass properties contains the following fields:

[Source Entry Type](#) Select the entry type that the pass should read from the identity store.

Note

This field is available/visible for provisioning jobs (action tasks) only.

[Retrieve Attributes From Pending Value](#) If the task operates on pending value object (see [Pending Value Objects \[page 95\]](#)) you can specify whether the pass should read the values from the pending value object or the parent entry. If the pass should update information about an entry in the target system, you keep this option deselected to retrieve the values from the attributes of the parent entry. To retrieve values from the pending value object (PVO), select this option.

To retrieve values from both the entry itself and the pending value object, you need to use a script to retrieve values from either the entry or the pending value object, depending on what you have selected here.

Note

This option also affects how the syntax for retrieving attributes from the referenced entries on the [Destination](#) tab works.

Note

This field is available/visible for provisioning jobs (action tasks) only.

[Identity Store](#) Select this option if the source of the pass is an identity store.

Note

If this option is selected, the fields [Database](#) and [Encrypt Connection String](#) are disabled.

Note

If identity store is selected as the source, the options for both defining a filter and a SQL query (fields [Filter](#) and [SQL Query](#)) are enabled.

[Identity Center Database](#) Select this option to use the Identity Center database as the source.

If this option is selected, the fields [Database](#) and [Encrypt Connection String](#), including the possibility to use a filter, are disabled. You are still able to define a SQL query.

[Database](#) Select this option to use other database (not the Identity Center database). It is normally a temporary database created by one of the From-passes, but could be any other data sources.

Enter the JDBC connection for the database. Choose ... to use the JDBC connection wizard and fill in the necessary parameters.

The possibility to use a filter is disabled if *Database* is selected as the source. You are still able to define a SQL query.

<i>Encrypt Connection String</i>	Select this option to specify that the connection string should be stored encrypted in the job configuration file. It will be displayed using asterisks in the field. This option is only enabled when the <i>Database</i> check box is selected.

<i>Filter</i> (Database Query)	Use the filter to define the data to be read from the source. The filter is only enabled when the <i>Identity Store</i> option is selected. Select the option and then enter the filter in the field that appears. You can use the filter editor to create the filter (the <i>Edit...</i> button just above the right upper corner of the field). The filter will automatically generate SQL query. To view the query, select the <i>SQL Query</i> option. You can enter the filter in lowercase and/or uppercase letters. The search is case insensitive.

Note

If you select the *SQL Query* option and then manually edit the query, the *Filter* option will be disabled. To activate *Filter* again, you must clear the SQL query field.

<i>Include Inactive Entries</i>	Only available if the <i>Filter</i> option is selected. Select this option to include the inactive entries when generating a filter and the SQL query.

<i>Use Attribute ID Instead of Attribute Name in SQL Query</i>	Only available if the <i>Filter</i> option is selected. The filter and the SQL query are made more generic by using the attribute names instead of attribute IDs. Select this option to use the attributes IDs and not the attribute names when generating a filter and the SQL query.

<i>SQL Query</i>	Enter the SQL query to define the data to be read from the source. The SQL query is only enabled when the <i>Identity Store</i> option is selected. Select the option and then enter the SQL in the field that appears. You may use the <i>Filter</i> option to help you generate an SQL query, that you so can extend. When filter editor is used to generate the SQL query, the search is always case insensitive.

Note

If you manually edit the SQL query, the *Filter* option will be disabled. To activate *Filter* again, you must clear the SQL query field.

→ Recommendation

If the SQL query refers to the current identity store, we recommend you to use the variable `<% $IDSID%>` to avoid possible issues in case of transport.

For example, if you refer to the ID of the current identity store with `mcidstore=1` or `is_id=1`, we recommend you to change it to `mcidstore=%$IDSID%` or respectively `is_id=%$IDSID%`.

📘 Note

When the result of an SQL query contains MSKEY column, you can expect the following:

- If the value in the MSKEY column can be mapped to the MSKEY of an existing entry of a given entry type, you will have this entry as an input on the [Destination](#) tab of the pass.
Only when the value in the MSKEY column is the MSKEY of an existing pending value object (PVO), SAP Identity Management checks further the value of the MX_ENTRY_REFERENCE attribute on this PVO. If the value of the attribute (used to hold the reference to the entry owning the PVO) is the MSKEY of an entry that can be mapped to an existing entry, then you will have this entry (the one owning the PVO) as an input on the [Destination](#) tab.
- If the value in the MSKEY column cannot be mapped to the MSKEY of an existing entry of a given entry type, you will have the result from the query as an input on the [Destination](#) tab of the pass.

Related Information

[Specifying the Source for To-Passes \[page 327\]](#)

9.1.12.3.10 Destination (To Identity Store Pass)

Gives an overview of the properties on the [Destination](#) tab of the To Identity store pass properties.

When configuring a To Identity store pass, you use the [Destination](#) tab to specify information about the identity store you are going to update. The [Destination](#) tab of the pass properties contains the following fields:

[Identity store](#)

The list contains all defined identity stores in the Identity Center. Select the identity store you want to update.

[Entry type](#)

Select the entry type that should be used when entries are written to the identity store. When you use a data destination template to populate the definitions table, only the attributes defined for the entry type are listed.

📘 Note

Selecting an entry type depends on the pass operation:

- If this pass creates entries in the identity store (add operation), defining the entry type in this field is mandatory. The created entries will be of this entry type.
- If this pass modifies or deletes entries in the identity store (changetype operation is modify or

delete), defining the entry type in this field is ignored. In this case, only the entry type of the entry's MSKEYVALUE in the definitions table is considered.

Multivalue delimiter

Select the character you want to use as a delimiter for multivalue attributes. The default value is the pipe character (|).

User info

Enter a text that will be written to each of the attributes in the identity store. This can, for instance, be used to specify who updated the value or add some information about the process which is executed. The default value of the field is **JobId=%\$ddm.mcjob%**, which corresponds to the MSKEY of the job, which contains the pass property.

The values of *User Info* field are not displayed in the Identity Management User Interface. They are recorded in the database in two places:

- In the *mcProcessInfo* column of *MXI_LINK* table.
- In the *UserID* column of *MXI_VALUES* table.

Invoking scripts from this field is not supported.

Definitions table

Use the definitions table to enter the definitions for the pass.

You can use a template to simplify the work with the pass definitions (*Insert template* button).

In the *Entry Type* field are displayed only the entry types of the identity store selected in the *Identity Store* field above. In case that this pass was imported as part of a package from another Identity Management system, note the following:

- If *Self* was selected in the *Identity Store* field in the source system, after the import to the target system, in the *Entry Type* field are available entry types of the identity store of the target system, in which the pass was imported.
- If an identity store, different than *Self*, was specified in the *Identity Store* field in the source system, and after the import into the target system there is an identity store with the same name, then in the *Entry Type* field are displayed the entry types of the identity store with this name in the target system.
- If an identity store, different than *Self*, was specified in the *Identity Store* field in the source system, and after the import into the target system there is no identity store with the same name, no entry types will be available in the *Entry Type* field on the target system.

Here is an example of the entry types available in the target system, depending on the configurations in the source system:

Source Identity Management System	Target Identity Management System
Identity Store: <i>Self</i> Entry Type: Entry types from the identity store schema of the current identity store on the source system	Identity Store: <i>Self</i> Entry Type: Entry types from the identity store schema of the current identity store on the target system
Identity Store: <i>IdentityStoreX</i> Entry Type: Entry types from the identity store schema of IdentityStoreX on the source system	Identity Store: <i>IdentityStoreX</i> (an identity store with this name exists on the target system) Entry Type: Entry types from the identity store schema of IdentityStoreX on the target system
Identity Store: <i>IdentityStoreX</i> Entry Type: Entry types from the identity store schema of IdentityStoreX on the source system	Identity Store: <i>IdentityStoreX</i> (an identity store with this name does not exist on the target system) Entry Type: No entry types available

Related Information

[Specifying the Destination for a To Identity Store Pass \[page 329\]](#)

9.1.12.3.11 Destination (To ASCII File Pass)

Gives an overview of the properties on the *Destination* tab of the To ASCII file pass properties.

When configuring a To ASCII file pass, you can use the *Destination* tab to enter information about the ASCII file that is generated with this pass. The *Destination* tab of the pass properties contains the following fields:

<i>File name</i>	Enter the name of the ASCII file you want to use as the output file in this pass. Choose the ... button to the right of the field if you want to browse for the file.
<i>Split line using</i>	Enter a character which is used to split one "row" in the data source into several rows in the output file. You must be sure that this character is used consistently within the data source to get a consistent result.
<i>Reset output file</i>	Select this option if you want to reset the ASCII file before the pass is executed.

<i>Binary output</i>	Select this option to specify that the file will be written using UTF-16.
<i>Character set</i>	Select one of the predefined character sets in the list: • Auto The runtime engine determines the character set based on the default character set specified in the Java environment. • ANSI (ISO-8859-1) • UTF-8
<i>Generate CSV file</i>	Select this option if you want to create a CSV (Comma Separated Values) file. All output fields are written to one line in the output file, using the CSV delimiter. Many applications are able to read and write CSV files.
<i>Include CSV header</i>	Select this option to write a CSV header to the file. This option is only enabled if Generate CSV file is enabled.
<i>Add quotes</i>	Select this option to add quotes around all attributes. This is used to preserve special characters in the attributes. This option is only enabled if Generate CSV file is enabled.
<i>CSV delimiter</i>	Enter the character you want to use as the CSV field delimiter. Use the keywords TAB and SPACE to insert a tab character or a space in the output file. The reason for adding a delimiter in a comma separated file is that different localizations of Microsoft Windows (and other applications) use different separators. In parts of Europe, the comma character is the decimal separator, and is therefore unsuitable as a separator. A semicolon is normally used instead. This option is only enabled if Generate CSV file is enabled.
<i>Default value</i>	Enter the value (text) you want to insert as a default value in any empty fields when generating the CSV file. For example, this is useful when updating a Microsoft Exchange server based on the CSV file. If you specify ~DEL in the Default value field, any empty fields in the source will then be replaced by ~DEL. When the Microsoft Exchange server is updated, the contents of the corresponding fields (attributes) will be removed. This option is only enabled if Generate CSV file is enabled.

Definitions table

Use the definitions table to enter the definitions for the pass.

You can use a template to simplify the work with the pass definitions (*Insert template* button).

Related Information

[Specifying the Destination for a To ASCII File Pass \[page 356\]](#)

9.1.12.3.12 Destination (To Database Pass)

Gives an overview of the properties on the *Destination* tab of the To Database pass properties.

When configuring a To Database pass, you can use the *Destination* tab to enter information about the database you want to update. The *Destination* tab of the pass properties contains the following fields:

<i>Database</i>	Specify the target data repository (enter an JDBC URL). You can use the JDBC connection wizard to create the URL. Choose ... to use the wizard and fill in the necessary parameters.

<i>Encrypt connection string</i>	Select this option to specify that the connection string should be stored encrypted in the job configuration file. It will be displayed using asterisks in the field.

<i>SQL updating</i>	This option determines what kind of definitions this pass contains. To use definitions where the key is the first line and other attributes follow on the next lines, leave this option empty. This is the default value. Select this option to use normal SQL statements such as SELECT, UPDATE, CREATE or DELETE. The first line of the definitions will not be executed. It is only used as a key in the delta database. Enabling this option will remove columns <i>Attribute</i> and <i>Data type</i> from the definitions table. Each line in the <i>Value</i> column contains one SQL statement.

Note

When using Microsoft SQL Server, it is recommended that all SELECT queries should include "with (nolock)". This will reduce the table locking, improve performance, and reduce the probability of deadlocks. For instance:

```
SELECT <columns> FROM <table> WITH (NOLOCK) WHERE <column>='<value>'
```

<i>Database table name</i>	Enter the name of the database table or equivalent.

Definitions table Use the definitions table to enter the definitions for the pass.
You can use a template to simplify the work with the pass definitions ([Insert template](#) button).

Note

All variables and constants are enclosed in a % pair (%name%). If there is a % sign in the output, you must use the keyword \$PERCENT. Likewise, if you need a | (pipe character) in the output, you use the keyword \$PIPE.

Related Information

[Specifying the Destination for a To Database Pass \[page 358\]](#)

9.1.12.3.13 Destination (To Generic Pass)

When configuring a To Generic pass, you can use the [Destination](#) tab to enter information about the data repository you want to update.

The [Destination](#) tab of the pass properties contains the following fields:

Open destination

The script is used to open the destination.

The script receives a collection with all attribute names defined in the [Definitions table](#) as the input parameter.

View and edit the script by choosing [Edit...](#) to the right of the fields. If no script is specified, choosing [Edit...](#) you can create a new script.

The script must return nothing if the processing was successful. If an error occurs during execution, the script must return a string containing the error message to be added to the log file, and the pass is aborted.

Next data entry

The script is used to read each entry in the temporary database and update the destination accordingly.

The script receives a collection with all attribute names defined in the [Definitions table](#) as the input parameter.

Note

If you are running on Microsoft SQL Server and the destination is the same as the source of the pass, the SELECT statement of the source must be on the form

```
SELECT with (nolock)
```

to avoid lock error during the update operation.

View and edit the script by choosing [Edit...](#) to the right of the fields. If no script is specified, choosing [Edit...](#) you can create a new script.

The script must return nothing if the processing was successful. If an error occurs during execution, the script must return a string containing the error message to be added to the log file, and the pass is aborted.

Close destination	The script is used to close the destination after the processing is finished. The script has no input parameters. View and edit the script by choosing Edit... to the right of the fields. If no script is specified, choosing Edit... you can create a new script. The script must return nothing if the processing was successful. If an error occurs during execution, the script must return a string containing the error message to be added to the log file, and the pass is aborted.

Definitions table	Use the definitions table to enter the definitions for the pass. You can use a template to simplify the work with the pass definitions (Insert template button).

Related Information

[Specifying the Destination for a To Generic Pass \[page 360\]](#)

9.1.12.3.14 Destination (To LDAP Directory Pass)

When configuring a To LDAP Directory pass, you can use the [Destination](#) tab to enter information about the directory you want to use.

Note

LDAP version 2 is the protocol for communication with the directory server. When using this protocol, only entries in Latin characters are read correctly. Make sure you do not use non-Latin characters as they are read incorrectly.

The [Destination](#) tab of the pass properties contains the following fields:

Directory computer name	Enter the name or address of the host running the LDAP server.

Directory LDAP port	Enter the port number. This is normally 389.

Directory login name	Enter the DN (distinguished name) used to log on to the directory server. This user must have update privileges.

Note

If you select *Kerberos* as the security option in the *Security option* field and the fields *Directory login name* and *Directory login password* are empty, the user will be authenticated using the current login credentials. This only applies to Active Directory or other directory servers supporting the Kerberos protocol.

Directory login password

Enter the correct login password (for the distinguished name you specified in the previous field).

Note

This parameter is stored as unencrypted in the .DSE file, and will be visible to anyone with the access to this file. Make sure that the file is access protected.

If you leave the field empty, anonymous login is used. This is, however, normally not sufficient for updating the data in the directory.

Max LDAP errors to tolerate

Enter the number of the LDAP errors the runtime engine should tolerate before canceling the job.

Character set

A character set that the LDAP server uses.

Security option

Use this list to specify the security level and protocol for communication with the directory server. You can select one of the following values:

- *Simple authentication*
The login name and login password are sent unencrypted to the directory server. This is the default mode of operation.
- *SSL*
If the directory server you are connecting to supports SSL (Secure Sockets Layer) communication, you can specify that the runtime engine should use this connection when reading or writing data. The default port number when using SSL is 636. Ensure that you specify the correct port number in the URL in the *LDAP URL* field.
Using SSL encrypts the communication channel to protect sensitive data. After the secure communication channel is set up, the login name and password are submitted to the server. For more information, see *SSL Communication Using the Java Runtime Engine* in *SAP Identity Management Security Guide*.
- *Kerberos*
This option is only relevant to communication with Active Directory servers or other directory servers supporting the Kerberos protocol. The current user's user name and password are then used to log on to the directory server. In this case, it is not necessary to fill in the fields *Directory login name* and *Directory login password*.

Add quotes

Select this option to add quotes around each part of the DN (distinguished name), except the top level. This may be necessary to handle special characters or national characters, for instance if there are characters that could be interpreted as delimiters.

Note

If the DN already contains any quotes, additional quotes will not be added.

Definitions table	Use the definitions table to enter the definitions for the pass. You can use a template to simplify the work with the pass definitions (<i>Insert template</i> button).

Related Information

[Specifying the Destination for a To LDAP Directory Pass \[page 361\]](#)

[SSL Communication Using the Java Runtime Engine](#)

9.1.12.3.15 Destination (To LDIF or DSML File Pass)

When configuring a To LDIF or DSML file pass, you can use the *Destination* tab to enter information about the LDIF file you want to update.

The *Destination* tab of the pass properties contains the following fields:

File name	Enter the name of the LDIF or the DSML file you want to use in this pass. Choose the ... button to the right of the field if you want to browse for the directory where you want to create the file.
Split line string	Enter a character which is used to split one "row" in the data source into several rows in the output file. You must be sure that this character is used consistently within the data source to get a consistent result.
Reset output file	Select this option if you want to reset (clear) the LDIF file before the pass is executed. If not, all the new data is appended to the existing data.
Generate DSML format	Select this option to create a file in Directory Services Markup Language (DSML) format.
Generate attributes for empty values	There may be attributes with no value (empty attributes) - for instance, if the value is deleted in the source of the pass. Select this option to specify that these attributes are to be written to the LDIF file with an empty value.
Add quotes	Select this option to add quotes around each part of the DN (distinguished name), except the top level. This may be necessary to handle special characters or national characters, for instance if there are characters that could be interpreted as delimiters.

Note

If the DN already contains any quotes, additional quotes will not be added.

Definitions Use the definitions table to enter the definitions for the pass.

table You can use a template to simplify the work with the pass definitions (*Insert template* button).

Related Information

[Specifying the Destination for a To LDIF or DSML File Pass \[page 363\]](#)

9.1.12.3.16 Destination (To Custom Pass)

When configuring a To Custom pass, you can use the *Destination* tab to enter the necessary parameters for your Java class.

The *Destination* tab of the pass properties contains the following fields:

Pass type Enter the name of the Java class to be called. The class must be accessible with the given CLASSPATH.

Parameter table Enter the parameter names and the corresponding values in the table. These parameters are submitted to the class when it is called.

Note

The parameter names are case-sensitive.

Definitions Use the definitions table to enter the definitions for the pass.

table You can use a template to simplify the work with the pass definitions (*Insert template* button).

Related Information

[Specifying the Destination for a To Custom Pass \[page 364\]](#)

9.1.12.3.17 Destination (Shell Execute Pass)

When configuring a Shell execute pass, you can use the [Destination](#) tab to enter information about the command you want to run.

The [Destination](#) tab of the pass properties contains the following fields:

[Hide application/shell command](#)

Select this option to specify that the window of the executing command will be hidden. In most cases, this option should be selected, because it is not necessary to see the external application when executing. However, during the development phase, it may be useful to see what is going on.

Note

Be careful selecting this option with [Wait for execution](#). If the application for any reason does not terminate as expected, you may need to close the application using Task Manager.

[Wait for execution](#)

Select this option if you want to wait until the task is finished before the job proceeds. For example, this is necessary if the job is used to extract information from an external data source. If sending data to a printer, it might not be necessary to wait for execution.

Note

Be careful selecting this option with [Hide application/shell command](#). If the application for any reason does not terminate as expected, you may need to close the application using Task Manager.

[Time limit](#)

Enter the maximum number of minutes you want to wait for the command to be executed before waiting is canceled and the job continues. If you enter 0 or leave the field empty, the runtime engine will wait for as long as it takes.

Note

Continuing the job may cause errors in subsequent passes, depending on the function of the Shell execute pass.

[Definitions table](#)

Use the definitions table to enter the definitions for the pass.

You can use a template to simplify the work with the pass definitions ([Insert template](#) button).

Related Information

[Specifying the Destination for a Shell Execute Pass \[page 366\]](#)

9.1.12.3.18 Delta (To-Passes)

Use the [Delta](#) tab to enter information about the delta for the passes.

The [Delta](#) tab in the properties of the To-passes contains the following fields:

Enable delta	Select this option to specify that you want to create a delta for this pass. If this option is selected, the fields SAP Identity Management Database or Database , and Delta identifier must be filled in.
SAP Identity Management database	Select this option to use the SAP Identity Management database as the delta database. If this option is selected, the fields Database and Encrypt connection string are disabled.
Database	Select this option to use other database (not the Identity Management database) as the delta database. Enter the JDBC connection for the database. Choose ... to use the JDBC connection wizard and fill in the necessary parameters.
Encrypt connection string	Select this option to specify that the connection string should be stored encrypted in the job configuration file. It will be displayed using asterisks in the field. This option is only enabled when the Database check box is selected.
Delta identifier	The delta database may contain delta information from more than one pass. This identifier is added to make the delta of each pass unique.
Delta key	The field contains the entries' key in the delta database. The field is used in two ways, depending on the pass type: • For To ASCII file passes, Shell execute passes and To Database passes where the option SQL updating in the Destination tab is selected, you enter the delta key here. Use the same syntax as when defining the delta key for other pass types. • For all other pass types, the delta key is specified as the first line of the pass definition (the naming attribute). The delta key is displayed here, but cannot be modified. For more information about the delta key and how the delta mechanism works when adding new role and privilege assignments for a user, see The Delta Key (To-Passes) [page 1073].
Source key	This is a unique reference to the entry in the data source. Enter the column in the source that contains the unique key as a recordset variable. You can combine several columns to create a unique key. You can also enter any text or reference any variable or constant.

If the data source is a database that the runtime engine is able to access, the context menu contains all columns in the data source.

<i>Generate delta only</i>	Select this option to produce the delta file only. No data is updated in the destination. This function can, for instance, be used for testing purposes, to regenerate a delta file or to produce a delta database for statistical purposes.
<i>Skip unchanged entries</i>	This option indicates whether to check for unchanged entries or not. If this option is not selected, all entries will be written to the target every time the job is executed. If it is selected, only the modified or new entries will be updated with the log information, although the delta will not be used when updating the target repository. This option is selected by default.
<i>Mark for deletion</i>	Select this option to specify that the runtime engine should mark entries as deleted. In the delta database, the operation code, fingerprint, etc. of each entry is updated when the job is run. New entries are added. To identify deleted entries, the delta database is searched to find missing entries and their operation code is set accordingly. As this operation can be time-consuming, you can turn it off if it is not necessary to identify deleted entries, for instance in a To ASCII file pass.
<i>Max limit for mark for deletion</i>	Enter the maximum percentage of entries that may be marked for deletion. If this limit is exceeded, nothing is marked for deletion in the delta database. This is to prevent deletions caused by errors in the input data.
<i>Max real updates</i>	Enter the number of entries that can be modified with this pass. If the limit is reached, the pass stops and the job continues with the other passes. This can be used to spread the load when performing large updates, for instance during three successive nights.

Note

Make sure this function is used together with a delta database that takes care of modification and that the data is not too dynamic in relation to the number of updates you specify.

<i>Audit trail level</i>	You can choose from three levels: • <i>None</i> No audit trail is made. • <i>Entry level</i> Information about each entry modified is stored in the delta database. • <i>Attribute level</i> Information about each modified attribute is logged.

Note

A detailed audit trail level may reduce performance and increase the size of the delta database.

<i>Max number of entries in</i>	You can control the size of the audit trail table by defining a maximum number of entries.

Note

If you leave the field empty or enter 0, the table size may increase dramatically.

Delete entries marked for delete in current job

Automatic deletion option that specifies that the runtime engine should delete the records that, according to the delta database, have been deleted.

Note that disabling this option automatically disables the [Delete entries marked for delete in earlier jobs](#) option as well.

Note

This option is not relevant for the following To-passes:

- To ASCII file pass
- To Database pass when the [SQL updating](#) option is enabled
- Shell execute pass

Delete entries marked for delete in earlier jobs

Automatic deletion option that specifies that the runtime engine should also delete records that a previous run of the job has flagged for deletion.

Note that enabling this option automatically enables the [Delete entries marked for delete in current job](#) option as well. You cannot enable this option alone.

Note

This option is not relevant for the following To-passes:

- To ASCII file pass
- To Database pass when the [SQL updating](#) option is enabled
- Shell execute pass

Autodelete script (To Generic pass only)

This script is used to delete entries in the data repository.

The script receives the DN from the delta database as the input parameter. It is called once for each entry fulfilling the criteria specified by the automatic deletion properties.

View and edit the script by choosing [Edit...](#) to the right of the fields. If no script is specified, choosing [Edit...](#) you can create a new script.

The script must return nothing if the processing was successful. If an error occurs during execution, the script must return a string containing the error message to be added to the log file, and the pass is aborted.

Related Information

[Configuring Delta for Passes \[page 379\]](#)

[The Delta Database \[page 1078\]](#)

9.1.12.3.19 The Delta Key (To-Passes)

For most pass types the first (blue) line of the pass definition (the naming attribute) is used to define the key in the delta database.

The delta key is displayed read-only in the *Delta key* field on the *Delta* tab. For the following pass types, the delta key is defined directly in the *Delta key* field on the *Delta* tab:

- To ASCII file pass
- Shell execute pass
- To Database passe with *SQL updating* option selected

Any unique attribute can be used as the delta key, such as the e-mail address or employee number. Note that a person's name is not normally unique.

If you want the delta mechanism to work when adding a number of new role and privilege assignments for a user, you can do the following:

1. Define a custom attribute to be used as the delta key, for example DELTA_KEY.
2. Configure the following pass definition on the *Destination* tab:

Attribute	Value
DELTA_KEY	Value1_Value2_Value3
MSKEYVALUE	Value1
MXREF_MX_ROLE	Value2
MXREF_MX_PRIVILEGE	Value3

3. Run the job.

Composite Delta Keys in a To Database Pass

In a To Database pass the delta key is normally a single unique attribute. But in some cases it may be necessary to use more than one attribute to create a unique reference (composite key). This is done using the + prefix.

For instance:

```
personname=%name%
```

```
+location=%location%
```

```
title=%title%
```

In this example the person name and location attributes together constitute the unique key.

When delta is enabled for this pass we have to change this to:

```
personname=%name%!%location%
```

```
+location=%location%
```

```
title=%title%
```

The key in the delta database will be %name%!!%location% (in expanded form), but when updating the destination database, only the part of the key field preceding the !! characters will be used for the person name attribute.

Note

When deleting entries based on the delta, the automatic deletion will not identify entries with a composite key. To delete entries, use the *Mark for deletion* option and create a separate pass to perform the deletion.

Composite Delta Keys in To LDAP Directory Passes

In a To LDAP directory pass, the distinguished name (DN) will be the unique key for the directory entry and also the delta key.

In a directory, composite keys are useful for group management. An entry can be a member of multiple groups and the combination of the group's distinguished name and the entry's distinguished name constitutes the unique key.

This can be defined in the following way:

```
dn=<group dn>!!<person dn>
```

```
objectclass=group
```

```
cn=%groupname%
```

The complete <group dn>!!<person dn> will be used as the delta key, but only <group dn> will be used when writing to the directory.

Related Information

[The Delta Key \(From-Passes\) \[page 1050\]](#)

[The Naming Attribute \[page 1083\]](#)

9.1.12.3.20 Prefix Options (To-Passes)

When you specify the definitions for a To-pass, the first column in the definitions table can contain a prefix that indicates special treatment of the following line.

There are also prefixes that can be used to specify the attribute operations. These prefixes are added to each value.

Prefixes Common to All To-Passes

<i>Except from delta database</i> (@)	Indicates that this attribute will not be "counted" when calculating the checksum in the delta operation. This prefix can be combined with other prefixes. The other prefixes will then be part of the <i>Attribute</i> column.
<i>Disable</i> (#)	Indicates that the line is a comment.

Prefixes Specific to To Identity Store Passes

<i>Write only when adding entry</i> (.)	Specifies that this attribute is written when an entry is added, but not when an entry is modified.

Prefixes Specific to To ASCII File Passes

If you are not creating a CSV file (i.e. the option *Generate CSV file* is not selected), you can add a header and/or trailer to the ASCII file you produce. This is useful, for instance, when generating HTML files.

<i>Header line</i> (<)	Indicates that the line is part of the header.

<i>Trailer line</i> (>)	Indicates that the line is part of the trailer.

Note

If you are using variable file names, it is not possible to have a trailer.

An example:

Prefix	Value
	Keystring
<	Header text 1
<	Header text 2
	Repeating field %Field1%

	Repeating field %Field2%
>	Trailer text 1
>	Trailer text 2

Prefixes Specific to To Database Passes

Primary key (+) Indicates that the current column will be part of the primary key in addition to the naming attribute (composite key).

For instance:

Name	%Name%
+Location	%Location%
Email	%Email%

In this case, *Location* is part of the primary key when writing entries to the database.

Note

See *The Delta Key (To-Passes)* for more information about using a composite key in a pass with delta enabled.

Note

A composite key can not be used together with CHANGETYPE Delete.

Prefixes Specific to To LDAP Directory Passes

Write only when adding entry (.) Specifies that this attribute is written when an LDAP add operation is performed, but not in the case of a modify operation. The objectclass attribute is a typical candidate for using this prefix, because many LDAP directories (such as Microsoft Exchange) do not allow modification of this attribute. A specification of this attribute would be:

```
.objectclass=top|remote|person
```

The prefix can also be used to specify that a password attribute should be written only when an entry is created.

```
.userpassword=unknown
```

<i>Add attribute value (+)</i>	This prefix can be used for multivalue attributes. A new value is added, without overwriting the existing value(s). If one or more of the values already exists, the operation fails.
<i>Delete attribute value (-)</i>	This prefix is used with multivalue attributes. It indicates that the specified value(s) are removed from the attribute. If one or more of the values does not exist, the operation fails. Note The meaning of this prefix changed from version 6.0 to 6.1.
<i>Replace attribute (~)</i>	This prefix indicates that the attribute is first deleted and then added. (This may be used in the circumstance that a normal "replace" fails.)
<i>Compare attribute (?)</i>	This prefix indicates that multivalue attributes are compared to the existing values. Only changed elements are deleted/added.

Prefixes Specific to To LDIF or DSML File Passes

<i>Write only when adding entry (.)</i>	Specifies that this attribute is written when an LDAP add operation is performed, but not in the case of a modify operation. The objectclass attribute is a typical candidate for using this prefix, because many LDAP directories (such as Microsoft Exchange) do not allow modification of this attribute. A specification of this attribute would be: <code>.objectclass=top remote person</code> The prefix can also be used to specify that a password attribute should be written only when an entry is created. <code>.userpassword=unknown</code>
<i>Add attribute value (+)</i>	This prefix can be used for multivalue attributes. A new value is added, without overwriting the existing value(s). If one or more of the values already exists, the operation fails.
<i>Delete attribute value (-)</i>	This prefix is used with multivalue attributes. It indicates that the specified value(s) are removed from the attribute. If one or more of the values does not exist, the operation fails. Note The meaning of this prefix changed from version 6.0 to 6.1.

9.1.12.4 The Delta Database

The delta database keeps track of which data records are modified. It will contain a hashed value of each record, which is used to check whether any fields have been updated.

The delta database tables are created with the installation of SAP Identity Management Core component. You can select it in the *Delta Database* field on the *Delta* tab in the pass definition.

The delta database contains four tables:

- [Delta_Defs Table \[page 1078\]](#)
- [Job_Execution Table \[page 1078\]](#)
- [Logentries Table \[page 1079\]](#)
- [AuditTrail Table \[page 1081\]](#)

9.1.12.4.1 Delta_Defs Table

The Delta_Defs table contains information about all delta identifiers specified in the pass properties.

The table has the following columns:

Column	Description
<i>OwnerID</i>	Unique identifier
<i>Name</i>	Delta identifier specified on the <i>Delta</i> tab in the pass properties.
<i>DELTAFIELDS</i>	Attributes listed on the <i>Destination</i> tab.

9.1.12.4.2 Job_Execution Table

The Job_Execution table contains information about the previous executions of the job. Each time a job is run, a new row is added to this table to update the job history.

The table has the following columns.

Column	Description
<i>ID</i>	Unique reference
<i>JobNumber</i>	Identifies a run of this job.
<i>OwnerID</i>	Reference to the OwnerID column in the Delta_Defs table.
<i>Machine</i>	Name of the computer that ran this execution of the job.

Column	Description
<i>Description</i>	Description of the job. The runtime engine will fill in a short description and version number.
<i>RunDate</i>	This is the date/time when the job started.
<i>Mods</i>	Number of modifications
<i>TimeUsed</i>	Time used for the job (in seconds).
<i>New</i>	Number of new entries
<i>Noop</i>	Number of no-operations
<i>Dels</i>	Number of deletes
<i>Recon</i>	Number of reconciliation transactions. Entries that are marked for refresh in this execution of the job. (For future use.)
<i>Errors</i>	Number of errors
<i>DNConflicts</i>	This indicates the number of duplicate DNs detected.

9.1.12.4.3 Logentries Table

The Logentries table contains the fingerprint information for the data.

The table has the following columns.

Column	Description
<i>DN</i>	Contains the distinguished name of the entry, which is used as a lookup in this table.
<i>OwnerID</i>	Reference to the OwnerID column in the Delta_Defs table
<i>FingerprINT</i>	Contains a fingerprint (hash value) of the entry. When a new job is executed, a fingerprint will be made of each entry, and compared to the fingerprint in this table.
<i>JOB</i>	Shows which job last updated this DN. This is a number, which refers to the job table.

Column	Description
<i>Operation</i>	This indicates which operation was performed on the entry. It may have one of the following values: 0: Conflict (conflicts are reported only when the Delta database is used) 1: Not changed 2: Entry modified 3: Entry added 4: Entry was deleted (marked for deletion) in current job 5: Entry was deleted (marked for deletion) in earlier job runs 6: Entry physically deleted. Can only occur using the <i>Automatic deletion</i> function in To-passes.
<i>Changejob</i>	Shows the job number of the job that performed the operation in the previous column.
<i>PrevOp</i>	When the Operation column is updated, the previous value is copied to this column.
<i>Errcode</i>	Errorcode 0 means OK. When a To LDAP from LDIF or a To LDAP directory pass is run, this column contains an LDAP error code. When a To database pass is run, the column contains an error code returned by the database system. You can use the value in this column to get a text description for an LDAP error if there is a field in the To-pass containing the LDAP error code. If the field is called Errcode (as in the delta database), you can get the error code by referencing the field in the following way: ##errcode% This can be done by defining a To ASCII file pass that runs after a To LDAP directory pass, where the data source is the delta database. The SQL statement could be: Select * from logentries where errcode >0 In the <i>Definitions</i> section you enter: Error message: ##errcode% Note It is not possible to use this method to get a description of a database error code.
<i>DeltaID</i>	Reference to the ID column in the Job_Execution table

Column	Description
<i>SOURCEKEY</i>	A unique reference to the entry in the data source
<i>EntID</i>	Sequence number used as a reference in the AuditTrail table.

9.1.12.4.4 AuditTrail Table

The AuditTrail table is used to store the necessary information for the audit trail. The table is empty if none of the passes in the job has activated the function.

The table has the following columns.

Column	Description
<i>RefID</i>	Reference to the EntID column in the Logentries table
<i>JobNumber</i>	Reference to the Changejob column in the Logentries table and the ID column in the Job_Execution table
<i>Operation</i>	This indicates which operation was performed on the entry. It may have one of the following values: 0: Conflict (conflicts are reported only when the Delta database is used) 1: Not changed 2: Entry modified 3: Entry added 4: Entry was deleted (marked for deletion) in current job 5: Entry was deleted (marked for deletion) in earlier job runs 6: Entry physically deleted. Can only occur using the <i>Automatic deletion</i> function in To-passes.
<i>SubOP</i>	The column contains 1 if an attribute (not entry) update was performed.

Column	Description
<i>ErrCode</i>	Errorcode 0 means OK. When a To LDAP from LDIF or a To LDAP directory pass is run, this column contains an LDAP error code. When a To database pass is run, the column contains an error code returned by the database system. You can use the value in this column to get a text description for an LDAP error if there is a field in the To-pass containing the LDAP error code. If the field is called Errcode (as in the delta database), you can get the error code by referencing the field in the following way: ##errcode% This can be done by defining a To ASCII file pass that runs after a To LDAP directory pass, where the data source is the delta database. The SQL statement could be: Select * from logentries where errcode >0 In the <i>Definitions</i> section you enter: Error message: ##errcode% Note It is not possible to use this method to get a description of a database error code.
<i>EntryData</i>	If you have selected generation of an audit trail on the attribute level, this column contains all current attribute values for this entry. Otherwise, the column is empty. When the entry is modified, the values of the original entry are removed, and the new values are inserted for the modified entry.
<i>Diffdata</i>	If you have selected generation of an audit trail on the attribute level, this column indicates the modifications made to the entry, using the following syntax: Code Syntax <pre><operation>:<attribute>:<old value>-><new value></pre> The operation may have one of the following values: - MOD: The attribute value has been modified. - DEL: The attribute value has been deleted. - NEW: The attribute value has been added.

Column	Description
Info	For future use

9.1.12.5 The Naming Attribute

The first line in the definitions table is used as a naming attribute. This attribute must have a unique value.

In a **To Identity Store pass**, the MSKEYVALUE is normally the naming attribute. Depending on whether the MSKEYVALUE is specified in the definitions table, as of version 8.0 SP07, Runtime patch level 17 and higher, you can expect the following:

- If the MSKEYVALUE is specified, when the pass is run, the MSKEYVALUE is taken into account regardless of its position in the table.
- If the MSKEYVALUE is not specified, when the pass is run, the first attribute, other than a comment or an empty value attribute, is treated as the MSKEYVALUE and is taken into account.

In a **directory**, the naming attribute is the DN attribute.

For a **database**, the naming attribute must be a unique attribute.

In a **To Generic pass** or **To Custom pass**, the naming attribute depends on the target that is to be updated. For other pass types there is no naming attribute.

The first line is also used to define the key in the delta database (From-passes/To-passes). The key is defined here, and the value is displayed read-only in the [Delta key](#) field on the [Delta](#) tab.

9.1.13 Constants (Job)

The Job Constant dialog box contains the following fields:

Field	Description
Name	Enter the name that you use when referencing the constant. Legal values are A-Z, 0-9 and _.

Field	Description
<i>Data Type</i>	Select a data type for the constant: • Boolean • Entry Reference • File Name • Folder Name • JDBC Driver • JDBC URL • Job Reference • LDAP URL • Password • Process Reference • String
<i>Value</i>	Enter the value of the constant. Note The value may also contain references to other (previously defined) constants.
<i>Encrypt Value</i>	Select this check box to specify that the value of the constant should be encrypted. It will then be hidden in the dialog box, and will be stored encrypted in the job configuration file or the Identity Management database. Note You can only encrypt values of the following data types: String, Password, JDBC URL, LDAP URL and OLE DB Connection String (valid for repository type constants only). Note If you encrypt a value, you will NOT be able turn it off again later. You need to delete the value, close the dialog box, reopen it and enter a new value.
<i>Description</i>	Enter a text describing the constant.

Related Information

[Adding or Modifying a Job Constant \[page 384\]](#)

9.1.14 Variables (Job)

The Job Variable dialog box contains the following fields:

Field	Description
<i>Name</i>	Enter the variable name that you use when referencing the variable. Legal values are A-Z, 0-9 and _.
<i>Data Type</i>	Select a data type for the constant: • Boolean • Entry Reference • File Name • Folder Name • JDBC Driver • JDBC URL • Job Reference • LDAP URL • Password • Process Reference • String
<i>Value</i>	Enter an initial value for the variable or leave this field empty. The value is updated by the scripts that use the variable.
<i>Encrypt Value</i>	Select this check box to specify that the value of the constant should be encrypted. It will then be hidden in the dialog box, and will be stored encrypted in the job configuration file or the Identity Management database. Note You can only encrypt values of the following data types: String, Password, JDBC URL, LDAP URL and OLE DB Connection String (valid for repository type constants only). Note If you encrypt a value, you will NOT be able turn it off again later. You need to delete the value, close the dialog box, reopen it and enter a new value.
<i>Description</i>	Enter a text describing the variable.

Related Information

[Adding or Modifying a Job Variable \[page 385\]](#)

9.1.15 Process Properties

A process defines a set of operations executed on entries that are stored in the identity store. The content of a process is defined by tasks, or other processes, which are executed in sequence.

The process properties contain the following tabs:

<i>General</i>	Enable/disable the process, define the process type or a repository for the process.
<i>Result Handling</i>	Configure result handling for the process.
<i>Documentation</i>	Contains a text field only, for the comments and the documentation (description) of the process.

Related Information

[General \(Process\) \[page 1086\]](#)

[Result Handling \(Process\) \[page 1089\]](#)

[Documentation \(Task/Process\) \[page 1112\]](#)

[Creating a Process \[page 220\]](#)

[Viewing and Configuring Process Properties \[page 221\]](#)

9.1.15.1 General (Process)

The *General* tab of the process properties is used to enable/disable the process, define the process type or a repository for the process.

The *General* tab of the process properties contains the following fields.

Field	Description
<i>Enabled</i>	Select this option to enable (or deselect to disable) the process.
<i>Process ID/Name</i>	The process ID is a number that identifies the process within the Identity Management database. You can modify the name of the process in the tree view.

Field	Description
	 Note Renaming of a public process is disabled.
<i>Qualified Name</i>	A qualified name is created and added to the process when it is made public. The qualified name is constructed in the following way: <package name>:process.<process name>. For example, <i>com.sap.MyPackage:process.MyProcess</i>.
<i>Process Type</i>	Define what type of process this process is. The following values are available: <i>Regular</i> A generic (unspecified) process. Default value for processes.  Note Regular processes can also be used as specific type of processes in different events, such as validation, execution, notification or result handling. For example, when you define an event process (Validate) on a repository type, role or privilege, instead of a Validate Add process, you can also use a Regular process. <i>Add Attribute Process</i> Add event process defined for an attribute. <i>Add Entry Process</i> Add event process defined for an entry type. <i>Add Member Process</i> Add member process for a repository, role or privilege. <i>Attestation Exception Process</i> Exception process for attestation. <i>Delete Attribute Process</i> Delete event process defined for an attribute. <i>Delete Entry Process</i> Delete event process defined for an entry type. <i>Deprovisioning Process</i> Deprovisioning process for a repository or privilege. <i>Form Action Process</i> A process that is referenced from a form. <i>Initialize Process</i> A process that is referenced from <i>Execute Process on Initialize</i> in any other processes. <i>Modify Attribute Process</i> Modify event process defined for an attribute. <i>Modify Entry Process</i> Modify event process defined for an entry type.

Field	Description
	• Modify Member Process Modify member process for a repository, role or privilege. • No Master Process A process defined on a repository or privilege, that is executed if a master privilege is missing. • Notification Process Notification process referenced from an approval or attestation task. • Notify Process Notification process defined for a repository, role or privilege. • Password Reset Error Process Error handling process for password reset. • Privilege Modify Process Modify event process for a repository or privilege. • Provisioning Process Provisioning process for a repository or privilege. • Remove Member Process Remove member process for a repository, role or privilege. • Request Complete Process Request complete process defined for an identity store. • Result Handling Failed Process Result handling process referenced from any other process or action task • Result Handling OK Process • Validate Add Process Validate add process for a repository, role or privilege. • Validate Modify Process Validate modify process for a repository, role or privilege. • Validate Remove Process Validate remove process for a repository, role or privilege.
Repository	You can select a repository that is linked to the process. When executing a process, the repository specified for the process is used. If no repository is specified, the repository specified for the parent process (a process that contains a link to the current process) will be used. If the process is initiated by an external call, the repository ID can include the repository ID as a parameter. If the process contains an action task, the repository specified for the process is used. You can not select a repository for an action task. The action task inherits the repository from the process.
Use Context Variables	Select this option to specify that context variables are to be created when this process is run.

Field	Description
	Unless you are using the context variable in the process hierarchy, you should keep this option deselected as it may have a negative impact on the performance.

Related Information

[Viewing and Configuring the General Properties for a Process \[page 222\]](#)

[Making a Process Public \[page 225\]](#)

9.1.15.2 Result Handling (Process)

Result handling of a process is configured on the [Result Handling](#) tab of the process properties.

The [Result Handling](#) tab of the process properties contains the following fields:

Execute Process on Initialize	Select an initialize process that should be started when this process is executed. You can select a process of type Regular or Initialize Process . Choose the ... button to the right of the field and then locate and select a process.
Execute Process on OK Result (Chain Result Actions)	The field displays the name of a process that should be executed if this process and all the sub-tasks or sub-processes have completed successfully. The process in the Execute Process on OK Result field is executed only for root level processes. If your process is not a root level process (that is, a sub-process), the chain process in the Execute Process on OK Result field of this sub-process will not be executed. However, if your root process contains sub-tasks, for example an action task, the no-chain process in the Execute Process on OK Result field of this action task will be executed after the action task is completed. You can select a process of type Regular or Result Handling OK Process. Choose the ... button to the right of the field and then locate and select a process.
Execute Process on Failed Result (Chain Result Actions)	The field displays the name of a process that should be executed if this process or any sub-task or sub-process has failed.

The process in the [Execute Process on Failed Result](#) field is executed only for root level processes. If your process is not a root level process (that is, a sub-process), the chain process in the [Execute Process on Failed Result](#) field of this sub-process will not be executed.

However, if your root process contains sub-tasks, for example an action task, the no-chain process in the [Execute Process on Failed Result](#) field of this action task will be executed after the action task is completed.

You can select a process of type [Regular](#) or [Result Handling Failed Process](#). Choose the button to the right of the field and then locate and select a process.

Note

If you export or copy a package, all references to processes which do not belong to the exported or copied package, will be lost.

Related Information

[Viewing and Configuring the Result Handling for a Process \[page 223\]](#)

9.1.15.3 Documentation (Task/Process)

Use the [Documentation](#) tab of the task or process properties to document a task/process.

The [Documentation](#) tab contains a text field where the documentation (e.g. description and other comments) can be entered.

Related Information

[Process Properties \[page 1086\]](#)

[Entering a Description of a Task \[page 268\]](#)

[Viewing and Entering the Description of a Process \[page 224\]](#)

9.1.16 Form Properties

Forms define the user interface which the end-users can use to modify or add data to the identity store.

The form properties contain the following tabs:

<i>General</i>	Configure the general properties for the form.
<i>Result Handling</i>	Configure result handling for the form.
<i>Attributes</i>	Define the attributes for the form.
<i>Parameters</i>	Configure the parameters for the guided activity forms, such as assignment request, view assignment request and password reset.
<i>Access Control</i>	Define the access control for the form.
<i>Presentation</i>	Configure the presentation of the form.
<i>Documentation</i>	Enter description of the form.

Related Information

[General \(Form\) \[page 1091\]](#)
[Result Handling \(Form\) \[page 1093\]](#)
[Attributes \(Form\) \[page 1094\]](#)
[Parameters \(Assignment Request Form\) \[page 1095\]](#)
[Parameters \(View Assignment Request Form\) \[page 1097\]](#)
[Parameters \(Password Reset Form\) \[page 1097\]](#)
[Access Control \(Form\) \[page 1100\]](#)
[Presentation \(Form\) \[page 1105\]](#)
[Documentation \(Form\) \[page 1108\]](#)

9.1.16.1 General (Form)

The *General* tab of the form properties contains the following fields.

Field	Description
<i>Enabled</i>	Select this option to enable (or deselect to disable) the form.

Field	Description
<i>FormID/Name</i>	The form ID is a number that identifies the form within the Identity Management database. You can modify the name of the form in the tree view.
<i>Form Type</i>	Define what type of form this form is. The following values are available: • <i>Regular</i> A generic (unspecified) form. • <i>Access Control Form</i> This form is used for attribute value validation. • <i>Display Form</i> The default display form for a given entry type. • <i>Search Form</i> The default search form for a given entry type.
<i>Repository</i>	You can select a repository that is linked to the form. When executing a form, the repository specified for the form is used.
<i>Use Context Variables</i>	Select this option to specify that context variables are to be created when this form is run. Unless you are using the context variable in the process hierarchy, you should keep this option deselected as it may have a negative impact on the performance.
<i>Use Inactive Entries</i>	Select this check box to specify that the form should include entries defined as <i>Inactive</i> in the identity store. These are entries that have the attribute MX_INACTIVE set.
<i>Extension Class</i>	If you have created an extension class using the Identity Center Extension Framework for input validation, you enter the name of the extension class you want to use. Select one of the check boxes <i>onLoad</i> or <i>onSubmit</i> to specify which methods in the class that you want to use. You have to select at least one of the check boxes to activate the extension class.

Field	Description
Custom Parameters	Enter custom parameters to the extension classes or form specific parameters. Choose Add. In the Custom Parameters dialog box, enter the following: • Property Enter the parameter name. • Type Select the data type of the parameter. • Value Depending on the selected data type, you are given different options when filling in the value of the parameter. Note Any name starting with MX_ is reserved for use by SAP Identity Management.

Related Information

[Configuring the General Properties of a Form \[page 276\]](#)

9.1.16.2 Result Handling (Form)

Result handling of a form is configured on the [Result Handling](#) tab of the form properties.

The [Result Handling](#) tab of the form properties contains the following fields:

Field	Description
Execute Process on Submit	The field displays the name of the process that should be executed if this form is submitted (saved) successfully. You can select a process of type Regular or Form Action Process. Choose the ... button to the right of the field and then locate and select the process.
Execute Process on OK Result	The field displays the name of the process that should be executed if this form completes successfully. You can select a process of type Regular or Result Handling OK Process. Choose the ... button to the right of the field and then locate and select the process.

Field	Description
Execute Process on Failed Result	The field displays the name of the process that should be executed if this form fails. You can select a process of type Regular or Result Handling Failed Process . Choose the ... button to the right of the field and then locate and select the process.

Note

If you export or copy a package, all references to processes which do not belong to the exported or copied package, will be lost. This is not valid for the process set to [Execute Process on Submit](#).

Related Information

[Configuring Result Handling for a Form \[page 277\]](#)

9.1.16.3 Attributes (Form)

Which attributes are displayed for a form in the Identity Management User Interface is configured on the [Attributes](#) tab of the form properties.

The [Attributes](#) tab contain the following fields:

Field	Description
Entry Type	Select the entry type for this form. Only the attributes that are allowed for this entry type will be listed in the attribute list. If there are mandatory attributes, they will be selected as Mandatory in the list. If you have selected Automatically create attributes for the identity store, any new attributes that are added to the entries written with the job, will be added to the entry type as well.  Note It will not be possible to write entries to the identity store without the attributes that are defined as mandatory for the entry type.

Field	Description
Report Form	Select this check box to specify that this is a form that is used to generate reports. (Not for conditional tasks, switch tasks and approval tasks)
This Form Creates a New Entry	Select This Form Creates a New Entry to specify that this form is used to create a new entry in the identity store. The attribute MSKEYVALUE must always contain a unique value. If MSKEYVALUE is not included or not entered, the value for MSKEYVALUE will be automatically generated. The only attribute that can be set to Read Only for such forms is MX_ENTRYTYPE.

Related Information

[Defining Attributes for a Form \[page 278\]](#)

9.1.16.4 Parameters (Assignment Request Form)

The specific parameters of an assignment request form are configured on the [Parameters](#) tab of the form properties.

The [Parameters](#) tab contains the following fields:

Field	Description
Entry Type	Select the entry type for this form. The form will only be available for entries of this entry type.
Context is Mandatory	Select this check box to specify that the assignment has to include a context. This is either selected by the user or provided as a default value for the assignment.
Context Type	The form can ask the user for a context for the assignment. You can choose between: None The user is not asked for context. User Select The form asks the user for a context type, and then a specific context of this type.

Field	Description
<i>Multiselect Context</i>	Select this check box to specify that the user is allowed to select more than one context for the assignment.
<i>Reference Type</i>	Select which entry type (role or privilege) that will be added by this form.
<i>Multiselect Reference</i>	Select this check box to specify that the user is allowed to select more than one role or privilege.
<i>Ask for Validity</i>	Select whether the form includes the fields to specify validity for the assignment. You can choose between: • <i>Never</i> The user is not asked for validity. • <i>Optional</i> The fields to specify validity are displayed, but the user is not required to fill them in.
<i>Ask for Reason</i>	The task can include the necessary field to specify a reason for the assignment. You can choose between: • <i>Never</i> The user is not asked for reason. • <i>Optional</i> The field to specify reason is displayed, but the user can leave the field empty. • <i>Mandatory</i> The user has to specify reason for the assignment request. Note It is not possible to modify the reason from the Identity Management User Interface. When modifying validity, the specified reason is the reason for the validity change.

Related Information

[Configuring Parameters for an Assignment Request Form \[page 283\]](#)

9.1.16.5 Parameters (View Assignment Request Form)

The specific parameters of a view assignment request form are configured on the [Parameters](#) tab of the form properties.

The [Parameters](#) tab contains the following fields:

Field	Description
Entry Type	Select the entry type for this form. The form will only be available for entries of this entry type.
Allow Restarting of Failed or Declined Privilege Assignments	Select this check box to specify that the form should include the option to restart failed or declined privilege assignments.
Allow Restarting of Failed Assignment Requests	Select this check box to specify that the form should allow the user to execute the form to restart failed assignments. This also includes any failed inherited assignments in the sub-tree. Any declined assignments will not be retried.

Related Information

[Configuring Parameters for a View Assignment Request Form \[page 284\]](#)

9.1.16.6 Parameters (Password Reset Form)

The specific parameters of a password reset form are configured on the [Parameters](#) tab of the form properties.

The [Parameters](#) tab contains the following fields:

Field	Description
Entry Type	Select the entry type for this form, usually MX_PERSON. The form will only be available for entries of this entry type.
Min no of Validation Answers	Specify how many of the defined questions the user has to answer. This number should be the same or higher than the number defined in the No of Answers Required field. If not, an error message is displayed and nothing is saved. This is used by any form where at least one of the authentication question attributes (MX_AUTHQ_nnn) is present.
Description	This field contains the caption for the first step in the password reset form. The default value is Provide Identification .

Field	Description
<i>Identification Attribute</i>	Select which attribute to use for unique user identification for password reset. You can choose between: • <i>MSKEYVALUE</i> This is the default. • <i>Use Alternative Attribute</i> Specify that you want to use the attribute selected in the <i>Alternative Identification Attr</i> field for unique identification instead of the MSKEYVALUE. This can for example be the e-mail address or employee number. Note You can choose an alternative identifier only among unique attributes. By default the only unique attribute is MSKEYVALUE. Thus, you should explicitly make an attribute unique, in order to see it as an alternative here. • <i>Use MSKEYVALUE and alt at.</i> Specify that you want to use both MSKEYVALUE and the alternative attribute for identification.
<i>Alternative Identification Attr</i>	If you have selected either <i>Use Alternative Attribute</i> or <i>Use MSKEYVALUE and alt at.</i> as <i>Identification Attribute</i>, you select the attribute that should be used for identification in the list.
<i>Description</i>	This field contains the caption for the second step in the password reset form. The default value is <i>Answer the Following Questions</i>.
<i>No of Questions to Show</i>	Specify the number of questions that will be displayed to verify the identity. The number should be the same or higher than the number defined in the <i>No of Answers Required</i> field, usually the same since there is no need of displaying more questions than the number required answering to. The questions will be picked from the authentication questions (MX_AUTHQ_XXX attributes) where the user has provided an answer.
<i>No of Answers Required</i>	Specify the number of answers that are required. The user should answer this number of questions. The number defined in this field should be the same or lower than the number defined in the <i>Min no of Validation Answers</i> field.
<i>Maximum no of Attempts</i>	Enter the maximum number of password reset attempts. There are three (3) attempts by default. If this number is exceeded, the user will not be allowed to reset the password.

Field	Description
<i>Description</i>	This field contains the caption for the third step in the password reset form. The default value is <i>Choose Finish to set the Password</i>.
<i>Password Creation Method</i>	Specify which method should be used to create the new password. You can choose between: <i>Automatically Generate</i> The password is automatically generated and should be sent to the user by SMS or e-mail, for example. In this case, a task which sends the new password to the user by SMS or e-mail needs to be created. Note In an organization using password provisioning, the users will have the same password both for the Identity Management User Interface and the e-mail system. In this case, it will be no point in sending the password to the user's e-mail address, as the user will not be able to access the e-mail system. <i>Ask the User</i> This is the default value. The user can enter a new password himself.
<i>Save the Password to UME</i>	Specify if you want to save the new password to UME. This is necessary to be able to log on to the Identity Management User Interface. Setting the productive password in UME is possible with one of the following SAP NetWeaver versions: - SAP NetWeaver 7.3 SP9 or higher - Enhancement Package 1 for SAP NetWeaver 7.3 SP6 Patch 3 or higher - SAP NetWeaver 7.4 SP2 or higher The option is enabled by default. The only use case where this option should be disabled is when the user does not have an account in UME, that is the user does not have access to Identity Management User Interface. What password policy is selected when creating a password, depends on whether this option is set or not. When the option is enabled, then the UME password policy is applied and not the Identity Management policy (even if defined). When the option is disabled, the defined Identity Management policy applies.

Field	Description
<i>Confirmation Message</i>	This field contains the confirmation for the password reset form. The default value is <i>Password Set</i> .
<i>Password Reset Failed Task</i>	Select the form that should be executed when the number of failed password reset attempts is exceeded.

Related Information

[Configuring Parameters for a Password Reset Form \[page 285\]](#)

[Setting Password Reset Parameters \[page 571\]](#)

9.1.16.7 Access Control (Form)

Who should be allowed to access the form from the Identity Management User Interface is configured on the *Access Control* tab of the form properties.

On the *Access Control* tab, choose *Add*. In the *Access Control* dialog box, the following fields appear:

Field	Description
Allow Access for	Select who this access control rule should apply to. You can choose between: • Anonymous The form is available to anyone accessing the Identity Management User Interface before logging on. • Logged-in user or identity store entry You can specify that the form should be available to all logged-in users or to one specific user. You can also limit the access to one specific identity store entry, such as a role or privilege. • Referral Specify that access is given through a referral via an attribute specified with the Referral attribute field. The form is available to all users who are referred to by the given referral attribute. • Filter Use a filter to specify to whom the form should be available. This can for example be used to give all users with a given location access to a form. Note This option is only available if you have defined filters in SAP Identity Management 7.2 and upgrade to version 8.0. For performance reasons we recommend that you do not use this option. You can use Logged-In User or Identity Store Entry and select a dynamic group instead.
Name	This field is enabled if you select Logged-in user or identity store entry in the Allow access for field. To give access to all logged-in users, leave the field empty. To give access to a specific user or identity store entry, enter an MSKEYVALUE from the identity store. To verify the name, choose Check Name. If only one entry is found, the value will be underlined. If more than one entry matches the value, a dialog box is displayed where you can select one of the entries. Only the 50 first entries are listed. If no entries match the criterion, a message is displayed and you must enter another value.

Field	Description
<i>Referral Attribute</i>	Select the referral attribute that should be used to give access to this form. A user referred to by this attribute is given access to execute the form on behalf of the users referring the given user via this referral attribute. This can for instance be a manager or a secretary who is given access to a form on behalf of a group of users.

Field	Description
<i>On Behalf of</i>	Select who the user should be able to execute the form on behalf of. You can choose between the following values: • <i>Everybody</i> The user is allowed to execute this form on behalf of everyone else. • <i>User or identity store entry</i> You can specify that users are allowed to execute this form on behalf of a specific entry. • <i>Relation - Self</i> The user is allowed to execute this form on behalf of himself. Self-service. • <i>Relation - Manager</i> The user is allowed to execute this form on behalf of all for whom she is defined as manager. • <i>Relation - Owner</i> This relation type is relevant for all entry types where an owner can be defined (roles, groups, dynamic groups). An entry can have several owners, and also a role can be defined as owner for an entry. All entries defined as owners, including members of any referenced roles, are allowed to execute this form on behalf of the entry. • <i>Relation - Role Manager</i> All entries defined as owners of the role are allowed to execute this form on behalf of all entries assigned to the role. • <i>Relation - Group Manager</i> All entries defined as owners of the group are allowed to execute this form on behalf of all entries assigned to the group. • <i>Relation - Dynamic Group Manager</i> All entries defined as owners of the dynamic group are allowed to execute this form on behalf of all entries assigned to the dynamic group. • <i>Relation - Privilege Manager</i> All entries defined as owners of the privilege are allowed to execute this form on behalf of all entries assigned to the privilege. • <i>Relation - Role Member</i> The user is allowed to execute this form on behalf of all members of the specified role. • <i>Relation - Dynamic Group Member</i> The user is allowed to execute this form on behalf of all members of the specified dynamic group.

Field	Description
	• Relation - Privilege Member The user is allowed to execute this form on behalf of all entries to which this privilege is assigned. • Relation - Group Member The user is allowed to execute this form on behalf of all members of the specified group. • Relation - Member of same role/privilege/group/dynamic group The user is allowed to execute this form on behalf of all members of the same role, privilege, group or dynamic group. • Relation - Anonymous The user is allowed to execute this form on behalf of not-logged in entries. You use the fields below to specify the details of the selected user type. Depending on what you select in this list, the corresponding field will be enabled.
Name	This field is enabled if you select User or identity store entry in the On Behalf of field. To specify that the user should be available for self service, leave the field empty. To specify that the user is allowed to execute this form on behalf of a specified user or identity store entry, enter an MSKEYVALUE from the identity store. To verify the name, choose Check Name. If only one entry is found, the value will be underlined. If more than one entry matches the value, a dialog box is displayed where you can select one of the entries. Only the 50 first entries are listed. If no entries match the criterion, a message is displayed and you must enter another value.

Field	Description
Filter	This field is enabled if you select Filter in the On Behalf offield. You can enter the filter in lowercase and/or uppercase letters. The search is case insensitive. Use the SQL query builder to create an SQL query that can be used to identify the users on behalf of whom the user should be allowed to execute this form according to this access control rule. You can use all fields in the identity store to build the criterion. If you enter the query manually, it is recommended to use one of the views installed with the product. If you later use the SQL query builder, the manually entered query will be overwritten. Note When using Microsoft SQL Server, it is recommended that all SELECT queries should include with (nolock). This will reduce the table locking, improve performance, and reduce the probability of deadlocks. For instance: <pre>SELECT <columns> FROM <table> WITH (NOLOCK) WHERE <column>='<value>'</pre>

Related Information

[Defining Form Access Control \[page 285\]](#)

9.1.16.8 Presentation (Form)

The Presentation of a form is configured on the [Presentation](#) tab of the form properties.

The [Presentation](#) tab of the form properties contains the following fields:

Field	Description
Display name	Enter a name for the form in the Identity Management User Interface. If no display name is specified, the form name is used as the display name.

Field	Description
Help URL	To be able to provide user assistance, you can enter the URL to a web based help system. If you fill in this field, a Help link is activated. When the user opens this link, the web page referenced with the Help URL is displayed. Note The help URL must be a valid HTTP URL.
Show Future Values	Select this check box to specify that you want to show a separate tab in the form with future values for this entry. Note This setting is ignored for display forms and search forms.
Show Assignment History	Select this check box to specify that you want to show a separate tab in the form with the assignment history for all assignments of this entry. Note This setting is ignored for display forms and search forms. Note Make sure to restrict access to the form only to those who should be able to see these values.
Show All Values	Select this check box to specify that you want to show a separate tab in the form with all values for this entry. Note This setting is ignored for display forms and search forms. Note Make sure to restrict access to the form only to those who should be able to see these values.

Field	Description
Show Historic Values	Historic values for all attributes of this entry can be listed on a separate tab in the form. Choose one of the following values: • None No historic values are displayed. This is the default value. • Selected Only the attributes defined in the Attributes list of the form are included in the list of historic values. • All All attributes of the entry are included in the list of historic values. Note This setting is ignored for display forms and search forms. Note Make sure to restrict access to the form only to those who should be able to see these values.
Task Header	Enter a text that is displayed as part of the form in the Identity Management User Interface. This text can be used to provide a description of the form to the user. Note For a guided activity the same text will be used for all steps in the activity. If you want or need to describe each step, this must be specified within the text itself. Note For a password reset form, the header is configured as part of the form parameters.
Task Description	Enter a text that is displayed in the list of self-service forms or as tool-tip in the form selector. Note This text is not used for a password reset form.

Field	Description
Submit Button	Select this check box to specify that the form should have a Submit button. You can also enter a custom caption. The default value is Save for a normal form. For a guided activity the text is Finish and cannot be changed.
Modify Button	Select this check box to specify that the form should have a Modify button. You can also enter a custom caption. The default value is Modify.
Confirmation Message	Enter a custom confirmation message that is displayed after choosing the Submit button.

Related Information

[Configuring the Presentation of a Form \[page 286\]](#)

[Viewing Attribute History \[page 551\]](#)

9.1.16.9 Documentation (Form)

Use the [Documentation](#) tab of the form properties to document a form.

The [Documentation](#) tab contains a text field where the documentation (e.g. description and other comments) can be entered.

Related Information

[Documenting a Form \[page 287\]](#)

9.1.17 Task Properties

A provisioning system in SAP Identity Management Developer Studio consists of processes and tasks. Provisioning is based on setting up a number of processes and the underlying tasks to perform the given actions.

There are different types of tasks:

Task	Description
Action Task	An action task is a task that references a job run when the task is executed. There is only one pass in the job, a To-pass. The source of the pass is an entry in the identity store that is ready to be processed by this task. The destination of the pass is the repository that is referenced in the task properties.
Wait Task	A wait task gives you a possibility to influence the task execution flow. There are no properties to configure for the wait task, as it always waits for all tasks before it to complete before the next task is executed.
Switch Task	A switch task controls task execution by evaluating the result of a condition and executing different task(s) depending on the result. The task has a number of case nodes that correspond to the possible values of the returned result. Add as many case nodes as needed to cover the different values returned from the condition. The result of the query or stored procedure is compared to the names of the case nodes. If a matching node is found, the tasks below this node are executed in the order they are defined below the node. If no node matches the result of the condition, the tasks under the <i>Else</i> node are executed, if such a node is defined.
Conditional Task	The conditional task has a <i>True</i> and a <i>False</i> node. The tasks under the <i>True</i> node are executed if the result of the test specified in the condition is true. The tasks under the <i>False</i> node are executed if the result of the test specified in the condition is false.

Task	Description
Approval Task	The approval task has three sub-nodes, Approve, Decline and Timeout. The tasks under the Approve node are executed when the request is approved. The tasks under the Decline node are executed if the request is declined. The tasks under the Timeout node are executed if a timeout occurs. For assignment approvals, the status of the approval is processed according to the value of the <code>MX_ATTR_STATE</code> attribute of the pending value object when the approval task completes (the whole task tree has been executed). You can use the notification process to notify the involved parties in the approval as the approval is processed. The process referenced in the field Execute Process on Initialize can be used to send an e-mail to the approvers. There is a job template that can be used to create the action task/job.
Attestation Task	Attestation tasks are used to perform periodic attestations of roles (privilege) assignments. You can use the notification process to send the notifications to the attester. There is a job template that can be used to create the action task/job.

Related Information

[Common Tabs \[page 1111\]](#)
[Conditional Task \[page 1112\]](#)
[Switch Task \[page 1115\]](#)
[Approval Task \[page 1117\]](#)
[Attestation Task \[page 1126\]](#)
[Action Task \[page 1129\]](#)
[Managing Process Contents \[page 225\]](#)
[Viewing and Configuring Task Properties \[page 238\]](#)
[Creating a Process \[page 220\]](#)

9.1.17.1 Common Tabs

Some tabs in the task properties are common for all tasks.

The common tabs are:

- [Result Handling](#) tab
- [Documentation](#) tab

Related Information

[Task Properties \[page 1109\]](#)

9.1.17.1.1 Result Handling (Tasks)

Result handling of a task is configured on the [Result Handling](#) tab of the properties.

The [Result Handling](#) tab of the task properties contains the following fields:

Execute Process on Initialize	Select an initialize process that should be started when this task is executed. To select a process, choose the ... button to the right of the field and then locate and select the process.
Execute Process on OK Result (Task Result Actions)	The field displays the name of the process that should be executed if the task completes successfully. To select a process, choose the ... button to the right of the field and then locate and select the process.
Execute Process on Failed Result (Task Result Actions)	The field displays the name of the process that should be executed if the task fails. To select a process, choose the ... button to the right of the field and then locate and select the process.

Note

If you export or copy a package, all references to processes which do not belong to the exported or copied package, will be lost.

Related Information

[Configuring Result Handling for a Task \[page 248\]](#)

9.1.17.1.2 Documentation (Task/Process)

Use the [Documentation](#) tab of the task or process properties to document a task/process.

The [Documentation](#) tab contains a text field where the documentation (e.g. description and other comments) can be entered.

Related Information

[Process Properties \[page 1086\]](#)

[Entering a Description of a Task \[page 268\]](#)

[Viewing and Entering the Description of a Process \[page 224\]](#)

9.1.17.2 Conditional Task

When using conditional tasks, different sub-tasks are executed depending on the result of a specific condition. The condition must always evaluate to true or false, and there is one task connected to each option.

The properties of a conditional task contain the following tabs:

General	Enable/disable the task, select a repository or add a condition. General (Conditional Task) [page 1113]
Result Handling	Define the result handling for the task. Result Handling (Tasks) [page 1111]
Documentation	Contains a text field only, for the comments and the documentation (description) of the task. Documentation (Task/Process) [page 1112]

Related Information

[Viewing and Configuring Conditional Task Properties \[page 241\]](#)

[Task Properties \[page 1109\]](#)

[Managing Process Contents \[page 225\]](#)

9.1.17.2.1 General (Conditional Task)

Use the [General](#) tab to enable/disable the task, define the repository or set the conditions for the conditional task.

The [General](#) tab of the conditional task properties contains the following fields:

Enabled	Select this option to enable the task (or deselect to disable the task). Only enabled tasks are included when running the provisioning configuration.
Task ID/Name	The task ID is a number that identifies the task within this Identity Management database. You can modify the task's name in the Process Flow Diagram.
Repository	You can select a repository that is linked to the task. When executing a task, the repository specified for the task is used. If no repository is specified, any repository specified for the parent task (any task above the current task in the tree) will be used. If the task is initiated by an external call, the repository ID can include the repository ID as a parameter.
Stored Procedure (Database Query)	To define conditions for a conditional task, you can use a stored procedure in the database that returns the result of a test. Select the option, and then select a stored procedure from the list that appears.
Filter (Database Query)	To define conditions for a conditional task, you can use a filter that returns a true/false result. Select the option and then enter the filter in the field that appears. You can use the filter editor to create the filter (the Edit... button just above the right upper corner of the field). The filter will automatically generate SQL query. To view the query, select the SQL Query option. You can enter the filter in lowercase and/or uppercase letters. The search is case insensitive. Note If you select the SQL Query option and then manually edit the query, the Filter option will be disabled. To activate Filter again, you must clear the SQL query field.
Include Inactive Entries	Only available if the Filter option is selected. Select this option to include the inactive entries when generating a filter and the SQL query.
Use Attribute ID Instead of Attribute Name in SQL Query	Only available if the Filter option is selected. The filter and the SQL query are made more generic by using the attribute names instead of attribute IDs. Select this option to use the attributes IDs and not the attribute names when generating a filter and the SQL query.

Note

The usage of attribute IDs in queries is not recommended if you plan to transport your changes to another system. You will have to adjust manually the attribute IDs in the query to match the attribute IDs in the target system.

[SQL Query](#) (Database Query)

To define conditions for a conditional task, you can use an SQL query that returns true/false result.

Select the option and then enter the SQL in the field that appears. You may use the [Filter](#) option to help you generate an SQL query, that you so can extend.

When filter editor is used to generate the SQL query, the search is always case insensitive.

Note

If you manually edit the SQL query, the [Filter](#) option will be disabled. To activate [Filter](#) again, you must clear the SQL query field.

Note

The query must use the Identity Management database, and the database user that is used by the runtime engine must have access to it.

→ Recommendation

If the SQL query refers to the current identity store, we recommend you to use the variable `<{$IDSID}>` to avoid possible issues in case of transport.

For example, if you refer to the ID of the current identity store with `mcidstore=1` or `is_id=1`, we recommend you to change it to `mcidstore=%{$IDSID%}` or respectively `is_id=%{$IDSID%}`.

Related Information

[Enabling or Disabling a Task \[page 245\]](#)

[Specifying a Repository for a Task \[page 247\]](#)

[Specifying the Conditions for a Conditional Task \[page 253\]](#)

9.1.17.3 Switch Task

A switch task controls task execution by evaluating the result of a condition and executing different task(s) depending on the result. The task has a number of case nodes that correspond to the possible values of the returned result.

The properties of a switch task contain the following tabs:

<i>General</i>	Enable/disable the task, select a repository or add a condition. General (Switch Task) [page 1115]
<i>Result Handling</i>	Define the result handling for the task. Result Handling (Tasks) [page 1111]
<i>Documentation</i>	Contains a text field only, for the comments and the documentation (description) of the task. Documentation (Task/Process) [page 1112]

Related Information

[Viewing and Configuring Switch Task Properties \[page 240\]](#)

[Task Properties \[page 1109\]](#)

[Managing Process Contents \[page 225\]](#)

9.1.17.3.1 General (Switch Task)

Use the *General* tab to enable/disable the task, define the repository or set the conditions for the switch task.

The *General* tab of the switch task properties contains the following fields:

<i>Enabled</i>	Select this option to enable the task (or deselect to disable the task). Only enabled tasks are included when running the provisioning configuration.
<i>Task ID/Name</i>	The task ID is a number that identifies the task within this Identity Management database. You can modify the task's name in the Process Flow Diagram.
<i>Repository</i>	You can select a repository that is linked to the task. When executing a task, the repository specified for the task is used. If no repository is specified, any repository specified for the parent task (any task above the current task in the tree) will be used.

	If the task is initiated by an external call, the repository ID can include the repository ID as a parameter.
<i>Stored Procedure</i> (Database Query)	To define conditions for a conditional task, you can use a stored procedure in the database that returns the result of a test. Select the option, and then select a stored procedure from the list that appears.
<i>Filter</i> (Database Query)	To define conditions for a switch task, you can use a filter that returns a value that will be used to select the corresponding case node. Select the option and then enter the filter in the field that appears. You can use the filter editor to create the filter (the <i>Edit...</i> button just above the right upper corner of the field). The filter will automatically generate SQL query. To view the query, select the <i>SQL Query</i> option. You can enter the filter in lowercase and/or uppercase letters. The search is case insensitive. Note If you select the <i>SQL Query</i> option and then manually edit the query, the <i>Filter</i> option will be disabled. To activate <i>Filter</i> again, you must clear the SQL query field.
<i>Include Inactive Entries</i>	Only available if the <i>Filter</i> option is selected. Select this option to include the inactive entries when generating a filter and the SQL query.
<i>Use Attribute ID Instead of Attribute Name in SQL Query</i>	Only available if the <i>Filter</i> option is selected. The filter and the SQL query are made more generic by using the attribute names instead of attribute IDs. Select this option to use the attributes IDs and not the attribute names when generating a filter and the SQL query. Note The usage of attribute IDs in queries is not recommended if you plan to transport you changes to another system. You will have to adjust manually the attribute IDs in the query to match the attribute IDs in the target system.
<i>Return Attribute</i>	Only available if the <i>Filter</i> option is selected. Select the attribute to be returned. Choose the ... button to the right of the field to browse for and select the attribute. The value is used to select the correct case node.
<i>SQL Query</i> (Database Query)	To define conditions for a switch task, you can use an SQL query that returns a value that will be used to select the corresponding case node. Select the option and then enter the SQL in the field that appears. You may use the <i>Filter</i> option to help you generate an SQL query, that you so can extend. When filter editor is used to generate the SQL query, the search is always case insensitive. Note If you manually edit the SQL query, the <i>Filter</i> option will be disabled. To activate <i>Filter</i> again, you must clear the SQL query field.

ⓘ Note

The query must use the Identity Management database, and the database user that is used by the runtime engine must have access to it.

→ Recommendation

If the SQL query refers to the current identity store, we recommend you to use the variable `<$$IDSID%` to avoid possible issues in case of transport.

For example, if you refer to the ID of the current identity store with `mcidstore=1` or `is_id=1`, we recommend you to change it to `mcidstore=%%$IDSID%` or respectively `is_id=%%$IDSID%`.

Related Information

[Enabling or Disabling a Task \[page 245\]](#)

[Specifying a Repository for a Task \[page 247\]](#)

[Specifying the Conditions for a Switch Task \[page 251\]](#)

9.1.17.4 Approval Task

Approval tasks are used to create a manual approval in a provisioning process. The task has three sub-nodes, [Approve](#), [Decline](#) and [Timeout](#). Depending on the result of the approval, the task(s) under one of these nodes are executed.

The properties of an approval task contain the following tabs:

[General](#)

Enable/disable the task or select a repository.

[General \(Approval Task\) \[page 1118\]](#)

[Result Handling](#)

Define the result handling for the task.

[Result Handling \(Tasks\) \[page 1111\]](#)

[Attributes](#)

ⓘ Note

The [Attributes](#) tab is only visible if the [Show Attributes](#) option is selected on the [Approval](#) tab of the approval task properties.

Define which attributes are displayed for this task in the Identity Management User Interface.

	Attributes (Approval Task) [page 1119]
<i>Approval</i>	Configure the details about the approval. Approval (Approval Task) [page 1119]
<i>Presentation</i>	Configure how the task is presented in the Identity Management User Interface. Presentation (Approval Task) [page 1125]
<i>Documentation</i>	Contains a text field only, for the comments and the documentation (description) of the task. Documentation (Task/Process) [page 1112]

Related Information

[Approval Processing \[page 34\]](#)

[Managing Process Contents \[page 225\]](#)

[Task Properties \[page 1109\]](#)

[Viewing and Configuring Approval Task Properties \[page 242\]](#)

9.1.17.4.1 General (Approval Task)

Use the *General* tab to enable/disable the task or define the repository for the approval task.

The *General* tab of the approval task properties contains the following fields.

Field	Description
<i>Enabled</i>	Select this option to enable the task (or deselect to disable the task). Only enabled tasks are included when running the provisioning configuration.
<i>Task ID/Name</i>	The task ID is a number that identifies the task within this Identity Management database. You can modify the task's name in the Process Flow Diagram.
<i>Repository</i>	You can select a repository that is linked to the task. When executing a task, the repository specified for the task is used. If no repository is specified, any repository specified for the parent task (any task above the current task in the tree) will be used. If the task is initiated by an external call, the repository ID can include the repository ID as a parameter.

Related Information

[Enabling or Disabling a Task \[page 245\]](#)

[Specifying a Repository for a Task \[page 247\]](#)

9.1.17.4.2 Attributes (Approval Task)

In the [Attributes](#) tab you can configure which attributes are displayed for the approval task in the Identity Management User Interface.

Note

The [Attributes](#) tab is only available if you have selected option [Show Attributes](#) on the [Approval](#) tab. The attributes you select here are displayed as part of the approval task to enable the approver to fill in additional fields when approving the request.

The [Attributes](#) tab of the approval task properties contains the following fields:

[Entry Type](#)

Select the entry type for this task. Only the attributes that are allowed for this entry type will be listed in the attribute list ([Task Attributes](#)) below. If there are mandatory attributes, they are selected as [Mandatory](#) in the list.

Note

It will not be possible to write entries to the identity store without the attributes that are defined as mandatory for the entry type.

[Task Attributes](#)

This list contains the attributes defined for the selected entry type in the [Entry Type](#) field.

Related Information

[Defining Attributes for the Identity Management User Interface for an Approval Task \[page 256\]](#)

9.1.17.4.3 Approval (Approval Task)

To configure the details about the approval, use the [Approval](#) tab of the approval task properties.

The [Approval](#) tab of the approval task properties contains the following fields:

[Approval Type](#)

The approval type determines how the approval task is configured and which fields and values are available.

You can choose between:

- [Assignment](#)
The task approves a role or a privilege assignment, and is started as a result of a member event role or privilege, for instance a [Validate add task](#).
- [Basic](#)
The task approves any other modification than assignments, for instance approving a change done in the Identity Management User Interface, by adding the approval task as a sub-task of a web-enabled task.

[Get Approvers from](#)

This list contains the possible sources where to find the approvers for the approval request.

Note

If you have chosen [Basic](#) as the approval type, the approvers are always defined on the approval task itself. See field [Approvals](#).

You can choose between:

- [Assignment Context](#)
If you are using context based assignment, the approvers can be defined on the context entry.
- [Pending Value Object](#)
This is the default value for an assignment approval (if the [Assignment](#) is selected as the approval type). When the event task is started, any approvers defined on the role/privilege are copied to the pending value object. You only need to select [Pending Value Object](#) if you want to modify the approvers manually before entering the approval task. Otherwise this is the same as choosing [Role/Privilege](#).
- [Role/Privilege](#)
The approvers are defined on the role or privilege.
- [Task](#)
The approvers are defined on the approval task itself. This is the only option available if [Basic](#) is selected as the approval type.

[Auto-Approve Inherited Assignments](#)

Select this option to specify that any inherited assignments are automatically approved or declined based on the action on the directly assigned role or privilege.

This will reduce the total number of approvals. If only direct assignments are to be approved, any indirect assignments as a result of the direct assignment will then automatically be approved when the direct assignment is approved.

Note

This option is only available when [Assignment](#) is selected as the approval type.

[Allow Validity Change](#)

Select this option to specify that the approver should be allowed to change the validity of the assignment as part of the approval process.

Note

This option is only available when [Assignment](#) is selected as the approval type.

Note

If the *Valid from* or *Valid to* date in the approval is modified by the approver to <Empty>, these dates are not stored as empty in database. They are stored as 1970-1-1 or 8099-12-30 respectively. But if the requester leaves them empty and the approver does not change them, they will be stored as empty.

Allow Manual Delegation

Select this option to specify that the manual delegation is enabled in the Identity Management User Interface.

Show Attributes

Select this option to enable adding attributes that could be filled in/modified when processing the approval. The *Attributes* tab is enabled and you can select the attributes to include in the approval task.

For basic approvals, you will be able to show and edit attributes of the entry being approved.

For assignment approvals, it is the pending value object which is viewed and edited. This means that you will need to have an action task which retrieves the updated from the pending value object, and performs the necessary actions based on the values.

Note

It is not possible to do any assignments in the approval task, so assignment attributes cannot be included in the list.

Re-Authenticate

To increase the security for handling approvals, you can specify that an approver has to prove his or her identity by providing the password when processing the approval. You can choose between the following values in the drop-down list:

- *None*
Never require re-authentication.
- *Approval*
Require a re-authentication when approving a request.
- *Always*
Require re-authentication when approving, declining, manually delegating or escalating an approval.

Allow Change of Appr.

Use this field to select who should be allowed to change the list of approvers with the internal function *UIS_Approval*. Possible values are:

- *None*
- *System*
- *System + Approvers*
- *System + Owners*
- *System + Approvers + Owners*

Reminder

Specify how long before the approval timeout a reminder should be sent to the approvers.

Select a unit from the list and enter a number to specify the value.

The default value is *0*, meaning no reminder is sent. If the *Timeout* is set to *Indefinite*, no reminder can be sent.

The reminder will be sent using the reminder message of the notification task (see field *Reminder Message*).

Timeout

Specify the timeout of the approval. The default value is indefinite (no timeout).

Select a unit from the list and enter a number to specify the timeout value. Additionally, you can choose *Escalation Rule* to specify which actions to take when the approval times out. It can either be declined or escalated.

Note

If, for any reason, the approval is not processed within 90 days from the initial request, it will be declined, regardless if the timeout specified here is longer than 90 days. To keep an approval active for a period longer than 90 days, you need to use the `MX_MAX_EXPIRY_TIMEOUT` global variable.

Expiry Timeout

`MX_MAX_EXPIRY_TIMEOUT` allows you to specify an expiry timeout of the approval task. The expiry timeout takes priority over the approval timeout. The global variable is supported for the 72 approval mechanism (as of 7.2 SP04 and higher) and can be used for basic and assignment approvals.

Note

Setting the global variable `MX_MAX_EXPIRY_TIMEOUT` affects only newly created approvals. All existing approvals and their expiry timeouts remain unchanged.

For assignment approvals, it works as follows:

- If expiry timeout is specified, it matches the Valid To date of the role/privilege assignment. If, for any reason, the approval is not processed from the date of the initial request to the Valid To date of the role/privilege assignment, the approval is declined.
- If the expiry timeout is reached but the approval timeout is still valid, the approval task takes the value of the expiry timeout and the approval is declined.
- If the approval timeout is reached but the expiry timeout is valid, the approval task takes the value of the expiry timeout and the approval is not declined.

Tip

Generally, if you want to keep an approval active up to 90 days, configure the approval timeout. If you want to keep an approval active longer than 90 days, configure the expiry timeout. For assignment approvals, always keep in mind that if the Valid To date of the role/privilege assignment is reached, the approval will be declined.

To specify an expiry timeout, you need to add `MX_MAX_EXPIRY_TIMEOUT` to the `MC_GLOBAL_VARIABLES` database table using the following query in all supported database systems:

```
INSERT INTO MC_GLOBAL_VARIABLES (VARNAME, VARVALUE, VARCATEGORY)
```



```
VALUES ( 'MX_MAX_EXPIRY_TIMEOUT', 'X', -2);
```

where 'X' is number of days in single quotes, for example: '100'.

Note

Make sure you specify the value in the correct format, otherwise, if specified incorrectly (for example: 'one hundred'), the default 90 days will be used for the expiry timeout of the approval task. However, when using MS SQL Server, Oracle and IBM DB2, you can omit the single quotes and specify the value as number of days only. For example: 100.

Escalation Rule

Specify which action to take when the specified timeout is reached or when an approval is escalated. You can choose between:

- [Decline](#)
- [Escalate to Manager](#) (available only when [Assignment](#) is selected as the approval type)
- [Escalate to New List](#)

You use the fields in the [Escalation](#) group to define the details of the escalations.

Required Approvals

Enter the number of required approvers to approve the request before it is considered approved. If this number of approvals is not reached until the defined timeout is reached, the approval is declined.

The default value is 1.

Note

If the number of approvers found for an assignment approval is lower than the required number of approvals, the approval request will be declined.

Insufficient Approvers

You can decide what should happen if the number of available approvers is less than the number of required approvers. This can happen in several situations, for instance if an approver is removed, or if the approver is a role, and the expanded role returns fewer users than the required number of approvers.

The following values are available:

- [Decline](#)
The approval will be declined. This is the default value.
- [Escalate](#)
The approval will be escalated, following the defined procedure for escalation.

Approvers

Note

Available only when [Basic](#) is selected as the approval type. When the approval type [Assignment](#) is selected, use the field [Get Approvers from](#) instead.

Add, modify or remove (using the buttons [Add](#), [Modify](#) and [Remove](#) respectively) the list of approvers with approval rules on the task.

Notification Task

Note

Before you can add a reference to the notification task, you must configure the notification templates and task.

Add a reference to the notification task (choose the ... button to the right of the field).

Initial Message

Select the message you want to use for the initial message that goes to the approvers.

Note

The option is available only when *Assignment* is selected as the approval type. When the approval type is *Basic*, the initial message selection is not possible.

Reminder Message

Select the message you want to use for the reminder message that is sent to the approvers.

Delegation Message

Select the message you want to use for delegation messages that are sent to the person to whom an approval is delegated (forwarded).

Escalation Message

Select the message you want to use for the message sent to the escalation approvers.

Completion Message

Select the message you want to use for the message sent to the requester and the user that is assigned the role/privilege.

Escalation Timeout

If you have defined *Escalate to Manager* or *Escalate to New List* in the *Escalation Rule* field, the approval will escalate if it times out without the required number of approvals.

The timeout for each escalation is defined on the approval task. This may be shorter than the timeout for the original approval. If the timeout is reached without the required number of approvals, it will escalate to the next level, until the value defined in *Max No. of Escalations* is reached or no approvers are found for a given level.

Select a unit from the list and enter a number to specify the timeout value.

Max No. of Escalations

If you have defined *Escalate to Manager* or *Escalate to New List* in the *Escalation Rule* field, the approval will escalate if it times out without the required number of approvals.

Specify how many times you want the approval to escalate before it is declined. The default value is *1*.

If no approvers are found for a given level during the escalation process, the approval will be declined, although the maximum number of escalations is not reached.

Escalation Approvers

If you have defined *Escalate to Manager* or *Escalate to New List* in the *Escalation Rule* field, the approval will escalate if it times out without the required number of approvals.

If *Escalate to New List* is selected in *Escalation Rule* and you have chosen *Task* as the source for the approvers, you specify the approvers for each escalation level in the list.

Enter the MSKEYVALUE of an entry in the identity store. This could be a user, a role or a privilege. To verify the name, choose *Check Name....* If only one entry is found, the value will be underlined. If more than one entry matches the value, a dialog box is displayed where you can select one of the

entries. Only the 50 first entries are listed. If no entries match the criterion, a message is displayed and you must enter another value.

Related Information

[Configuring the Approval \[page 257\]](#)

[Configuring the Approvers of an Assignment Approval \[page 259\]](#)

[Configuring the Approvers of a Basic Approval \[page 261\]](#)

[Adding or Modifying an Approval Rule for Basic Approvals \[page 262\]](#)

9.1.17.4.4 Presentation (Approval Task)

Configure how the task is presented in the Identity Management User Interface in the [Presentation](#) tab of the approval task properties.

The [Presentation](#) tab of the approval task properties contains the following fields:

Display Name	Enter a name for the task in the Identity Management User Interface. If no display name is specified, the task name is used as the display name.
Help URL	To be able to provide user assistance, you can enter the URL to a web based help system. If you fill in this field, a Help link is activated. When the user opens this link, the web page referenced with the Help URL is displayed.  Note The help URL must be a valid HTTP URL.
Task Header	Enter a text that is displayed as part of the task in the Identity Management User Interface. This text can be used to provide a description of the task to the user.
Task Description	Enter a text that is displayed in the list of self-service tasks or as tool-tip in the task selector.
Decline Button	An approval task has a Decline button. The Decline button can be disabled by deselecting the Decline Button option.
Confirmation Message	This field contains a custom confirmation message displayed after choosing the Approve button.

Related Information

[Configuring the Presentation of the Approval Task in the Identity Management User Interface \[page 266\]](#)

9.1.17.5 Attestation Task

Attestation tasks are used to perform periodic attestations of role (privilege) assignments.

The properties of an attestation task contain the following tabs:

<i>General</i>	Enable/disable the task or select a repository. General (Attestation Task) [page 1126]
<i>Result Handling</i>	Define the result handling for the task. Result Handling (Tasks) [page 1111]
<i>Attestation</i>	Configure the details about the attestation. Attestation (Attestation Task) [page 1127]
<i>Documentation</i>	Contains a text field only, for the comments and the documentation (description) of the task. Documentation (Task/Process) [page 1112]

Related Information

[Viewing and Configuring Attestation Task Properties \[page 243\]](#)

[Task Properties \[page 1109\]](#)

[Managing Process Contents \[page 225\]](#)

9.1.17.5.1 General (Attestation Task)

Use the *General* tab to enable/disable the task or define the repository for the attestation task.

The *General* tab of the attestation task properties contains the following fields.

Field	Description
<i>Enabled</i>	Select this option to enable the task (or deselect to disable the task). Only enabled tasks are included when running the provisioning configuration.
<i>Task ID/Name</i>	The task ID is a number that identifies the task within this Identity Management database. You can modify the task's name in the Process Flow Diagram.
<i>Repository</i>	You can select a repository that is linked to the task.

Field	Description
	When executing a task, the repository specified for the task is used. If no repository is specified, any repository specified for the parent task (any task above the current task in the tree) will be used. If the task is initiated by an external call, the repository ID can include the repository ID as a parameter.

Related Information

[Enabling or Disabling a Task \[page 245\]](#)

[Specifying a Repository for a Task \[page 247\]](#)

9.1.17.5.2 Attestation (Attestation Task)

Use the [Attestation](#) tab of the attestation task properties to configure details about the attestation.

The [Attestation](#) tab of the attestation task properties contains the following fields:

Get Attesters from	This list contains the possible sources where to find the attesters for the attestation request. You can choose between: • Task The attesters are defined in the Attester field below (that is, on the attestation task itself). • Role/Privilege The attesters are defined on the role or privilege with the attribute MX_ATTTESTER. To define the attesters, proceed as follows: 1. Log on to SAP Identity Management User Interface. 2. Select a role and choose the Modify Role Details or form. For privileges, choose the Modify Privilege Details form. 3. Select the Attestation tab and specify the attesters. • Manager The attester(s) will be the manager(s) defined on the user(s) with the attribute MX_MANAGER. • User Defined The attesters are set with context variables with the preprocessing task.
Attester	If you have chosen Task in Get Attesters from field, you enter the MSKEYVALUE of the attester. Note This must be a single user, not a role/privilege.

<i>Re-Authenticate</i>	Select this option to specify that an attester has to prove his or her identity by providing the password when processing the attestation.
<i>Allow Manual Delegation</i>	Select this option to specify that manual delegation is enabled in the Identity Management User Interface.
<i>Timeout</i>	Specify the timeout. The default value is indefinite (no timeout). Select a unit from the list and enter a number to specify the timeout value. Note If, for any reason, the attestation is not processed within 90 days from the initial request, it will be declined, regardless if the timeout specified here is longer than 90 days.
<i>Exception Task</i>	The exception task is used to process rejected attestation requests. For instance the task can be used to deassign any rejected assignments. It is called for each attestation request, not waiting for all attestation requests for the role/privilege. Note Make sure to select the <i>Use Context Variables</i> option on the exception task (on the <i>General</i> tab) to specify that the context variables are created.
<i>Notification Task</i>	Note Before you can add a reference to the notification task, you must configure the notification templates and task. Add a reference to the notification task (choose the ... button to the right of the field).
<i>Initial Message</i>	Select the message you want to use for the initial message that goes to the attester(s).
<i>Delegation Message</i>	Select the message you want to use for delegation messages that are sent to the person to whom an attestation request is delegated (forwarded).
<i>Reminder Message</i>	Select the message you want to use for the reminder message that is sent to the attester(s).
<i>Reminder</i>	Specify how long before the defined timeout a reminder (message) should be sent to the attester(s). Select a unit from the list and enter a number to specify the value.

Related Information

[Configuring the Attestation \[page 267\]](#)

9.1.17.6 Action Task

An action task is a task that references a job run when the task is executed.

There is only one pass in the job, a To-pass. The source of the pass is an entry in the identity store that is ready to be processed by this task.

The properties of an action task contain the following tabs:

<i>General</i>	Enable/disable the task. General (Action Task) [page 1129]
<i>Result Handling</i>	Define the result handling for the task. Result Handling (Tasks) [page 1111]
<i>Logging</i>	Specify information about a job's logging. Logging (Job/Action Task) [page 1024]
<i>State</i>	Shows information about the job's state. State (Job/Action Task) [page 1025]
<i>Documentation</i>	Contains a text field only, for the comments and the documentation (description) of the task. Documentation (Task/Process) [page 1112]

Related Information

[Task Properties \[page 1109\]](#)

[Viewing and Configuring Action Task Properties \[page 239\]](#)

[Managing Process Contents \[page 225\]](#)

9.1.17.6.1 General (Action Task)

Use the *General* tab to configure general options of the action task.

The *General* tab of the action task properties contains the following fields.

<i>Enabled</i>	Select this option to enable the task (or deselect to disable the task). Only enabled tasks are included when running the provisioning configuration.
<i>Task ID/Name</i>	The task ID is a number that identifies the task within this Identity Management database. You can modify the task's name in the Process Flow Diagram.

<i>Job ID/Name</i>	Displays the unique ID and the name of the job. You change name of the job in the tree view.
<i>Repository</i>	This field is visible only if you have upgraded from version 7.2, where you have specified a repository for an action task. As of version 8.0, repositories can be specified for processes. You can do the following: - Remove the repository from this action task. In this case, when you execute the action task, the repository specified for the process that contains the action task is used. Note Once the repository is removed, it is no longer possible to specify any repository for the action task. - Keep the repository of this action task. In this case, when you execute the action task, the repository specified for the task is used. If the task is initiated by an external call, the repository ID can include the repository ID as a parameter.
<i>Allow Parallel Provisioning</i>	Select this option to specify that this action task is enabled for parallel provisioning. Note Make sure you understand the concept before you enable parallel provisioning.
<i>Run by Dispatcher(s)</i>	Select the dispatcher(s) that are allowed to run this job.
<i>No. of Retries</i>	Specify how many times the task will be attempted again if it fails the first time. If this number is set to 0, it will not be re-tried.
<i>Retry Delay</i>	Select a unit in the list and enter a number to specify the period between each attempt to execute the task.
<i>Right Trim Source Attributes</i>	Select this option to specify that you want to remove any trailing spaces from attribute values read from the source files/repositories of the passes in the job.

Related Information

[Specifying the General Parameters of an Action Task \[page 244\]](#)

[Enabling or Disabling a Task \[page 245\]](#)

9.1.18 Internal Functions

When you edit the scripts, you can call internal functions to simplify your work.

When using the Java runtime engine, the script language is JavaScript, which is very similar to JScript.

Note

JScript is case sensitive, which means that you must call the internal functions using exactly the same casing as described.

Function	Description
uAddAuditInfo [page 1138]	Adds information to the detailed audit that is displayed on the "Monitoring" tab in the Identity Management User Interface.
uAppContextVar [page 1139]	Appends a value to a context variable.
uApplyPending [page 1140]	Applies a pending value.
uCheckFile [page 1141]	Returns the number of bytes in a file.
uCheckMembership [page 1141]	Checks if a given entry is a member of another entry.
uChecksumFile [page 1142]	Returns the checksum of a file.
uCkSum [page 1143]	Produces a simple checksum from a string.
uCompareHash [page 1143]	Compares a string previously hashed string. This can be used for authentication.
uDecrypt [page 1146]	Decrypts a string using the default encryption algorithm.
uDESDecrypt [page 1147]	Decrypts the given string using triple DES.
uDESEncrypt [page 1148]	Encrypts the given string using triple DES.
uEncrypt [page 1149]	Encrypts a string using the default encryption algorithm.
uError [page 1151]	Writes an error message to the log.
uErrMsg [page 1151]	Writes a message to the error log.
uExpandString [page 1152]	Expands occurrences of %\$constant% and %attribute% in a string.
uFingerPrint [page 1153]	Computes fingerprint for the given entry, using the supplied delta info.
uFromBase64 [page 1153]	Converts a string from Base64 to normal text.
uFromFile [page 1154]	Reads data from a file.
uFromHex [page 1155]	Decodes from hex to normal text.
uGeneratePassword [page 1156]	Generates a random password.
uGenHash [page 1156]	Generates a hashed string.
uGetApprovers [page 1158]	Returns a dictionary of possible approvers for given task and MSKey (Entry ID).
uGetAttrsOfType [page 1159]	Returns identity store attributes of the specified type.
uGetAuditID [page 1159]	Returns the current audit ID.
uGetChangeValues [page 1160]	Returns the old and new value when an attribute is changed.
uGetConstant [page 1160]	Returns the value of a constant (package constant, system constant or repository constant).
uGetContextVar [page 1161]	Returns the value of a given context variable.
uGetCount [page 1162]	Returns the number of entries processed in the current pass.
uGetDeltaInfo [page 1162]	Returns the values of OwnerID and DeltaFields for a given delta identifier.
uGetEntryID [page 1163]	Returns the ID of the current entry in the identity store.

Function	Description
uGetErrCount [page 1163]	Gets the number of errors that have occurred so far.
uGetErrorInfo [page 1164]	Returns info about the last error that occurred. This function is intended to be called from the error handler script.
uGetGeneralizedTimeZ [page 1165]	Returns the system time relative to GMT.
uGetGlobalVar [page 1165]	Returns the value of a global variable.
uGetGUID [page 1166]	Returns an unique string.
uGetIDStore [page 1166]	Returns the ID of the identity store if it is called from a provisioning job.
uGetIniFile [page 1167]	Reads an entry in an .INI file.
uGetPackageTaskID [page 1167]	Returns the ID of the task based on PackageID and TaskName.
uGetPassDocumentation [page 1168]	Returns the description (documentation) of a pass.
uGetPassSubject [page 1168]	Returns the name of the pass.
uGetProvisionUser [page 1169]	Returns the UserID info from an entry in the provisioning audit log.
uGetRepositoryID [page 1169]	Returns the repository ID of the current repository.
uGetRepositoryName [page 1169]	Returns the name of the current/specified repository.
uGetRepositoryPackageID [page 1170]	Returns the ID of the package that repository type of current repository belongs to.
uGetRepositoryVar [page 1170]	Returns the value of a repository variable.
uGetTaskID [page 1171]	Returns the task ID of the current task.
uGetTaskName [page 1171]	Returns the task name for a given task.
uGetTaskRoot [page 1172]	Returns the ID of the task that initiated the current provisioning (root task).
uGetUserVar [page 1172]	Returns the value of a job variable.
uGetWarnCount [page 1172]	Gets the number of warnings that have occurred so far.
uHexToByte [page 1173]	Decodes a hex value and returns byte[[]].
uHTMLEscape [page 1173]	Replaces UNICODE characters in a string with the corresponding HTML escape characters.
uHTMLUnescape [page 1174]	Replaces HTML escape characters with UNICODE characters.
uIncContextVar [page 1174]	Increments a context variable by 1.
uInfo [page 1175]	Writes an info message to the log.
uInt8ToDate [page 1175]	Converts data from Integer8 to a date format.
uIS_Approval [page 1176]	Processes an approval.
uIS_Compare [page 1178]	Compares all types of attributes to a specified value.
uIS_Create [page 1179]	Creates an entry in an identity store.
uIS_Delete [page 1179]	Deletes an attribute from an entry in the identity store.
uIS_Get [page 1180]	Returns an entry from the identity store based on the value of an attribute.

Function	Description
uIS_GetEntry [page 1180]	Returns all parameters from an entry (MSKEY).
uIS_GetError [page 1181]	Returns the error description according to the error number.
uIS_GetValue [page 1181]	Returns the value of a given attribute for a given entry in the identity store.
uIS_LinkEntries [page 1182]	This function is used to re-create links between sub-entries and parents in the identity store based on information read from a directory server (or other source).
uIS_nGetValues [page 1183]	Returns the values of a given attribute for a given entry in the identity store.
uIS_PrivReconcile [page 1184]	Reconciles the entries in an identity store with the defined roles and privileges.
uIS_RepairEntry [page 1185]	Repairs a given entry in the identity store by reconciling and rebuilding all assignments for this entry.
uIS_ResolveDynamicGroup [page 1186]	Resolves a dynamic group.
uIS_Revert [page 1186]	Sets an attribute or entry in the identity store to a given time/date.
uIS_SetApproval [page 1187]	Creates an approval for a given entry.
uIS_SetDirty [page 1188]	Sets a given entry (or its members) "dirty" to be picked up by the scheduled procedure for reconciliation.
uIS_SetValue [page 1189]	Updates the value of a given attribute for a given entry in the identity store.
uIS_sGet [page 1190]	Returns an entry from the identity store based on the value of a given attribute.
uIS_sGetValue [page 1191]	Returns the current value of an attribute in the identity store, or the previous value of the same attribute.
uIS_Touch [page 1191]	(Re)calculates the expiry time for an attribute in the identity store.
uIsRunning [page 1192]	Detects whether the runtime engine is running a full job or inserting a data source template.
uKillFiles [page 1192]	Deletes files.
uLDAPGetDN [page 1193]	Returns the DN returned by the specified LDAP URL.
uLDAPGetEntry [page 1194]	Returns the entry returned by the specified LDAP URL.
uLocaleToSapLanguage [page 1195]	Converts a Locale string code to a two-letter language code.
uLookupClose [page 1196]	Closes the active lookup database.
uLookupCreateDB [page 1196]	Creates a new lookup database.
uLookupDelete [page 1197]	Deletes an entry in the lookup database.
uLookupOpen [page 1197]	Opens a lookup database.
uLookupSeek [page 1198]	Searches a lookup database and returns a lookup value corresponding to the input parameter.
uMD5 [page 1199]	MD5-encodes a string. The string result will always be a 32-byte string.
uMGetContextVar [page 1199]	Returns all context variables matching the given filter as a Java properties map.
uNewEntry [page 1200]	Creates a new DSEEntry when returning the result of a From Generic pass.
uPreparedSelect [page 1201]	Returns the result of a Select statement in the Identity Management database.

Function	Description
uProvision [page 1203]	Initiates a provisioning task for a given entry in the identity store.
uPutIniFile [page 1204]	Updates an entry in an .INI file.
uReFormatDate [page 1205]	Converts a date to a string using the given format.
uReplaceBinary [page 1205]	"Replace" function for a string. The same as uReplaceString, but case sensitive.
uReplaceString [page 1206]	"Replace" function for a string ignoring case in matching.
uRepositoryConfig [page 1206]	Sets the repository state.
uRepositoryStatus [page 1207]	Checks repository active state.
uResetDelta [page 1207]	Resets delta information for a given OwnerID in the delta database.
uResolveDGMembership [page 1208]	Checks whether a given entry is a member of a dynamic group.
uRetryPrivilegeAdd [page 1209]	Retries a privilege assignment.
uRetryPrivilegeRemove [page 1209]	Retries a privilege removal.
uRunJobNow [page 1210]	Sets an Identity Center job ready to run.
uSapLang1ToLanguage [page 1211]	Converts a SAP one-character language code to a two-letter language code.
uSapLanguageToLang1 [page 1211]	Converts a two-letter language code to a SAP one-character language code.
uSapLanguageToLocale [page 1212]	Converts a two-letter language code to a Locale string code.
uSelect [page 1213]	Returns the result of a Select statement in the Identity database.
uSendSMTPMessage [page 1219]	Sends an SMTP message to the given recipient(s).
uSendSNMPTrap [page 1221]	Sends the specified SNMP trap message to the specified SNMP manager.
uSetContextVar [page 1221]	Updates the value of a context variable.
uSetExitcode [page 1222]	Sets an error level before the runtime engine exits.
uSetGlobalVar [page 1222]	Updates the value of a global variable.
uSetJobToError [page 1223]	Sets an Identity job to error state permanently.
uSetRepositoryVar [page 1223]	Updates the value of a repository variable.
uSetUserVar [page 1223]	Updates the value of a job variable.
uSHA1 [page 1224]	Returns the SHA1 checksum for a string.
uShell [page 1225]	Executes a shell command.
uShellRead [page 1225]	Executes shell command and reads output.
uSkip [page 1226]	Skips a record or skips current pass.
uSleep [page 1227]	Stops the execution of the job for the specified number of milliseconds.
uSplitString [page 1227]	Returns a Vector of strings.
uStop [page 1228]	Stops execution of a job.

Function	Description
uStrConv [page 1228]	Converts a string to lower case, upper case or mixed case.
uToBase64 [page 1229]	Converts a string to Base64.
uToBinFile [page 1230]	Writes binary data to a file, for instance pictures or certificates.
uToFile [page 1231]	Writes to a file.
uToHex [page 1232]	Converts a string to Hex codes.
uWarning [page 1233]	Writes a warning to the log.

VBScript specific functions

These are some commonly used VBScript function. Corresponding functions also exist in the Java version:

Function	Description
Instr [page 1135]	Finds a string within a string
Left [page 1136]	Returns characters from the left part of a string
Len [page 1136]	Returns the length of a string
Mid [page 1137]	Returns characters from the middle of a string
Right [page 1137]	Returns characters from the right of a string

9.1.18.1 Instr

Description	Finds a string within another string. This is a VB script function, which is very useful in creating user functions. This function is often used in combination with the functions Mid, Left and Right.	
Syntax	Java: iPos=Instr(Int Start, String InString, String SearchFor, int iCompare);	
Parameters	iStart	Start position to search in InString. 1 is the first character in InString.
	InString	String to search.
	SearchFor	String to search for in InString.
	iCompare	0: Case sensitive 1: Case insensitive
	Ipos	If >0, this is the position in InString where the SearchFor parameter is found. If <0, it means that SearchFor was not found.

Example: Java

```
var instring = "Madame";
var searchfor = "ad";
var start = 1;
var compare = 1; // Case is not significant
var pos = Instr(start, instring, searchfor, compare);

// pos should now be 2

searchfor = "ed";
pos = Instr(start, instring, searchfor, compare);

// pos should now be -1
```

9.1.18.2 Left

Description:	Returns the left part of a string.	
Syntax	OutString=Left(String InString, Int iLen);	
Parameters	InString	Original string.
	iLen	Number of characters to return.
	OutString	Returned string.

Example

```
InString = "Madame"
iLen = 4

OutString = Left(InString,iLen)
' --- OutString = "Mada"
```

9.1.18.3 Len

Description:	Returns the length of string.	
Syntax:	Length=Len(String InString);	
Parameters:	InString	Original string.
	Length	Number of characters in InString.

Example

```
InString = "Madame"  
Length = len(InString)  
' --- Length = 6
```

9.1.18.4 Mid

Description: Returns a substring of a string

Syntax: OutString=Mid(String InString, Int iStart, Int iLen);

Parameters:	InString	Original string.
	iStart	First position to extract.
	iLen	Number of characters to extract.
	OutString	Returned string.

Example

```
InString = "Madame"  
Istart = 3  
ILen = 2  
OutString = Mid(InString,iStart,iLen)  
' --- Outstring = "da"
```

9.1.18.5 Right

Description: Returns the right part of a string.

Syntax: OutString=Right(String InString, Int iLenFromRight);

Parameters:	InString	Original string.
	iLenFromRight	Length from right of string.
	OutString	Returned string.

Example

```
InString = "Madame"  
iLen = len("Madame")-3  
OutString = Right(InString,iLen)  
' --- OutString = "ame"
```

9.1.18.6 uAddAuditInfo

Description:	Adds information to the detailed audit that is displayed on the Monitoring tab in the Identity Management User Interface. For more information about the provisioning audit, see the <i>SAP Identity Management Solution Operation Guide</i> .	
Syntax:	OutString = uAddAuditInfo(int AuditID, int MSKey, int OtherMSKey, int LogLevel, int MsgType, String Title, String ShortText, String AdditionalInfo, int AdditionalInfoType);	
Parameters:	AuditID	ID of audit which this audit information belongs to.
	MSKey	This is always the MSKEY of the entry for which this audit information belongs. This will normally be the user. If AuditID is set, this should be the same MSKEY as in the mxp_audit table.
	OtherMSKey	In the case where a link is established, this is the MSKEY of the entry being referenced. This can link to any entry type. This will typically be the role or privilege which were assigned to the user.
	LogLevel	This is a numeric value, which can be used for filtering messages. The package constant AUDIT_EXTRA_LOGLEVEL is used to control which levels are logged. Only messages with a higher or equal level than this constant are logged. 0 No logging.
	MsgType	Message type can be one of the following: -1: Debug 0: Info 1: Warning 2: Error 3: Fatal
	Title	Title to show in tables.
	ShortText	Short description.
	Additional Info	Additional information.

Additional InfoType	This indicates the type of content in AddInfo. The following values are defined:	
	0:	Text
	1:	XML
	The logging application must set this according to the content	
OutString	Empty string, or error message prefixed with !ERROR:.	

Example

```
ThisAuditId = uGetAuditID();
// UserMSKEY = MSKEY of TestUser
// RoleMSKEY = MSKEY of TestRole
LogLevel = 1;
MsgType = 0; // Info
Title = "Assignment";
ShortText = "TestRole is assigned to TestUser";
AddInfo = "";
AddInfoType = 0; // Text
Res = uAddAuditInfo(ThisAuditID,
UserMSKEY,
RoleMSKEY,
LogLevel,
MsgType,
Title,
ShortText,
AddInfo,
AddInfoType);
```

Related Information

[SAP Identity Management Solution Operation Guide](#)

9.1.18.7 uAppContextVar

Description: Appends a string to an existing value of a context variable. If there is no existing value, the string will be the only value of the context variable. See also [uSetContextVar](#), [uGetContextVar](#) and [uIncContextVar](#).

Syntax: OutString=uAppContextVar(String Key, String Value);

Parameters: **Key** The name of the context variable.

Value The value to be appended.

OutString The new value, or error message prefixed with !ERROR:.

Example

```
uSetContextVar("MyContextVar", "ThisValue");
NewValue=uAppContextVar(MyContextVar, "ThatValue");

'The new value will now be ThisValueThatValue
```

Related Information

[Context Variables \[page 62\]](#)

[uSetContextVar \[page 1221\]](#)

[uGetContextVar \[page 1161\]](#)

[uIncContextVar \[page 1174\]](#)

9.1.18.8 uApplyPending

Description: Applies a pending value. It is only valid for entries of the type MX_PENDING_VALUE, and will only apply the value stored in the pending value object to the entries to which it belongs. See also the document *SAP Identity Management Identity Center: Implementing pending approvals* available on the SDN.

Syntax: Java: Status=uApplyPending(Int PendingMSKey, Int Flag);

Parameters: PendingMSKey MSKey (Entry ID) of the entry for which to apply pending operations.
ey

Flag	The following values are defined: 1: If this flag is set, any "valid from" is ignored and the pending value is applied anyway.

Status	Returns one of the following values: 0: OK -1: Not applied, validFrom is not yet pending -2: No such MSKey.

Example

```
mskey = Par.get("MSKEY");
OutString = uApplyPending(mskey, 0);
'OutString = 0
```

9.1.18.9 uCheckFile

Description: Returns the number of bytes in the file. 0 is returned if the file does not exist.

Syntax: iBytes=uCheckFile(String Filename);

Parameters: **Filename** File to check.

iBytes Number of bytes.

Example

```
FileName = "%$ddm.ddmpath%\checkfile.txt"
iBytes=uCheckFile(FileName)
' --- iBytes = <number of bytes>
```

9.1.18.10 uCheckMembership

Description: Checks if a given entry (child) is a direct member of another entry (parent) using the attribute MXREF_<ENTRYTYPE> attribute. See the document *Identity Store Schema* for details.

Syntax: Result=uCheckMembership(Int ParentMSKey, Int ChildMSKey);

Parameters: **ParentEntryID** MSKey of the parent entry.

ChildEntryID MSKey of the child entry.

Result Can be one of the following values:

< 0: Error

0: Not member

1: Member

Example

```
RoleMSKey=62
PersonMSKey=114
Result = uCheckMembership(RoleMSKey, PersonMSKey);
'Result = 1
```

The test finds that the person identified by MSKey 114 is a direct member of the role identified by 62.

Related Information

[Identity Store Schema](#)

9.1.18.11 uChecksumFile

Description: Returns the checksum of a file using MD5, SHA-1 or CRC32.

Syntax: CheckSum=uChecksumFile(String Algorithm, String FileName);

Parameters: Algori Specifies which algorithm to use. Possible values are:

thm
MD5
SHA-1
CRC32

FileNa The name of the file to be processed. The file name must include a valid path according to the
me operating system.

CheckS The returned checksum.
um

Example

```
// Main function: CheckSum  
  
function CheckSum(Par)  
{  
    //Compute a CRC32 checksum of the file given as parameter  
    // The result will be a string containing an 8 digit HEX number eg.  
    "1992b032"  
    return uChecksumFile("crc32",Par);  
}
```

9.1.18.12 uCkSum

Description:	Produces a simple checksum from a string. Returns the checksum of the string into the iLong variable. Note that this function may return the same number for different strings. Use the uMD5 function to compute a unique key from a string.	
Syntax:	iLong=uCkSum(String StringIn[, String CharacterEncoding]);	
Parameters:	StringIn	Original string.
	CharacterEncoding	Optional. Specifies which character set to use when calculating the checksum, and must be the name of a supported character set. Which character sets are supported varies from platform to platform, but every implementation of the Java platform is required to support the following standard character sets: • US-ASCII • ISO-8859-1 • UTF-8 • UTF-16BE • UTF-16LE • UTF-16 Other character sets may be available, but this requires a manual installation. If this parameter is not specified, the default character set for the platform is used.
	iLong	Returned checksum.

Example

```
StringIn = "Madame"  
iLong = uCkSum(StringIn)  
' --- iLong = <checksum>
```

9.1.18.13 uCompareHash

Description:	Compares a string previously hashed string. This can be used for authentication.

Note

If the value you are comparing to is written by the Identity Management User Interface, it was converted to uppercase before it is hashed. Make sure that the same attributes are handled consistently both when written by a job or from the Identity Management User Interface.

Syntax:	OutString=uCompareHash(String ClearTextValue, String CompareHashedValue[, String CharacterEncoding]);	
Parameters:	ClearTextValue	Clear text value to be compared.
	CompareHashedValue	Hashed value to compare with, typically generated with uGenHash .
	CharacterEncoding	(Optional) Specifies which character set to be used for encoding of the string. The value of the parameter must be the name of a supported character set. Which character sets are supported varies from platform to platform, but every implementation of the Java platform is required to support the following standard character sets: • US-ASCII • ISO-8859-1 • UTF-8 • UTF-16BE • UTF-16LE • UTF-16 Other character sets may be available, but this requires a manual installation. If this parameter is not specified, the default character set for the platform is used.
	OutString	Returns one of the following values: True - if the strings match False - if the strings do not match

Example

In this example, we use the default hashing algorithm:

```
String ClearText;
String HashedValue;

ClearText = "MyPassword";

HashedValue=uGenHash(ClearText);
if uCompareHash(ClearText, HashedValue)
/* Compare OK */
```

In this example we specify which hashing algorithm to use:

```
String ClearText;
String HashedValue;
String Algorithm;

ClearText = "MyPassword";
Algorithm = "SHA";

HashedValue=uGenHash(ClearText, Algorithm);
if uCompareHash(ClearText, HashedValue)
/* Compare OK */
```

Related Information

[Encryption \[page 68\]](#)

9.1.18.14 uCreateReport

Description:	Generates a <code>zip</code> file in hex format. This function is used to create standard reports in zip files containing <code>CSV</code> files.	
Syntax:	<code>uCreateReport(sql, reportName[, String CharacterEncoding]);</code>	
Parameters:	<code>sql</code>	The SQL select query for a given report.
	 Note On Microsoft SQL Server, the select queries should include "with (nolock)" or %NOLOCK% variable.	
	<code>reportName</code>	The name of the report file.
	<code>CharacterEncoding</code>	(Optional) Specifies which character set to be used for encoding of the report file. The value of the parameter must be the name of a supported character set. Which character sets are supported varies from platform to platform, but every implementation of the Java platform is required to support the following standard character sets: • US-ASCII • ISO-8859-1 • UTF-8 • UTF-16BE • UTF-16LE • UTF-16 Other character sets may be available, but this requires a manual installation. If this parameter is not specified, the default character set for the platform is used.

Related Information

[Standard Reports \[page 780\]](#)

9.1.18.15 uDecrypt

Description: Decrypts a string using the default encryption algorithm.

Syntax: OutString=uDecrypt(String Cipher[, String Key[, String CharacterEncoding]]);

Parameters:	Cipher	Encrypted string to decrypt. This is encrypted with uEncrypt . For details about the syntax of the encrypted string, see <i>SAP Identity Management Security Guide</i> .
	Key	Encryption key. This is mandatory if a user defined key was used in uEncrypt . If a default key is defined in the cipher, this parameters is ignored.
	CharacterEncoding	Optional. Specifies which character set to use when converting the decoded bytes to the returned string, and must be the name of a supported character set. Which character sets are supported varies from platform to platform, but every implementation of the Java platform is required to support the following standard character sets: • US-ASCII • ISO-8859-1 • UTF-8 • UTF-16BE • UTF-16LE • UTF-16 Other character sets may be available, but this requires a manual installation. If this parameter is not specified, the default character set for the platform is used.
	OutString	Clear text string or "!ERROR" if unable to decrypt.

Example

In this example we use the default encryption algorithm:

```
String ClearText;
String Cipher;
String Decrypted;

ClearText = "Text to be encrypted";

Cipher    = uEncrypt(ClearText);
Decrypted = uDecrypt(Cipher);
' --- Decrypted = "Text to be encrypted"
```

In this example we use a custom algorithm and key:

```
String ClearText;
String Cipher;
String Decrypted;
String Key;
String Algorithm;

ClearText = "Text to be encrypted";
```



```

Key      = "qwertyuiopasdfghjkljsdf"; /* Key must be exactly 24 bytes. */
Algorithm = "DES3CBC";

Cipher   = uEncrypt(ClearText, Algorithm, Key);
Decrypted = uDecrypt(Cipher, Key);
' --- Decrypted = "Text to be encrypted"

```

Related Information

[Encryption \[page 68\]](#)

[SAP Identity Management Security Guide](#)

[uEncrypt \[page 1149\]](#)

9.1.18.16 uDESDecrypt

Note

As of SAP NetWeaver Identity Management Identity Center 7.2, it is recommended to use the function [uDecrypt](#) instead.

Description: Decrypts the given string using triple DES. The key must be exactly 24 characters.

If an error occurs, an exception will be thrown.

Syntax: DecryptedString = uDESDecrypt(String EncryptionKey, String CipherText [, boolean encodeashex]);

Parameters:

EncryptionKey	The key used to decrypt the string. Can be hex-encoded, prefixed with {HEX}. If this is empty, the default key file is used. If this is not found, the internal key is used. It can contain a valid key file name, including the path. It can contain a valid key that is used for decryption.

CipherText	The string that is to be decrypted. Can be hex-encoded, prefixed with {HEX}.

encodeashex	Optional. False if the resulting should be returned as an unencoded string.

DecryptedString	The resulting decrypted string. Unless the optional 3rd parameter is set to false, it will be hex-encoded and prefixed with {HEX}.

Example

```
// Main function: Test
function Test(Par){
    // Using the internal encryption key, or one on file, if defined
    cryptPassword = uDESEncrypt("", "password");
    UserFunc.uErrMsg(1,"cryptPassword : " + cryptPassword);
    unCrypt = UserFunc.uDESDecrypt("", cryptPassword);
    UserFunc.uErrMsg(1,"UNCRYPT: " + unCrypt);
    // Using an explicitly given DES3 key
    cryptPassword2 = uDESEncrypt("qwertyuiopasdfghjklksdfr", "password"); /* Key
must be exactly 24 bytes. */
    UserFunc.uErrMsg(1,"cryptPassword2 : " + cryptPassword2);
    unCrypt2 = UserFunc.uDESDecrypt("qwertyuiopasdfghjklksdfr", cryptPassword2);
    UserFunc.uErrMsg(1,"UNCRYPT2: " + unCrypt2);
}
```

Related Information

[uDecrypt \[page 1146\]](#)

9.1.18.17 uDESEncrypt

Note

As of SAP NetWeaver Identity Management Identity Center 7.2, it is recommended to use the function [uEncrypt](#) instead.

Description: Encrypts the given string using triple DES. The key must be exactly 24 characters. The result is hex encoded and returned without any prefix. If an error occurs, an exception will be thrown.

Syntax: EncryptedString = uDESEncrypt(String EncryptionKey, String ClearText);

Parameters:

EncryptionKey	The encryption key. Can be hex-encoded, prefixed with {HEX}. If this is empty, the default key file is used. If this is not found, the internal key is used. It can contain a valid key file name, including the path. It can contain a valid key that is used for encryption.

ClearText	The string that is to be encrypted.

EncryptedString	The resulting encrypted string. It will be hex-encoded and prefixed with {HEX}.

Example

```
// Main function: Test
function Test(Par){
    // Using the internal encryption key, or one on file, if defined
    cryptPassword = uDESEncrypt("", "password");
    UserFunc.uErrMsg(1,"cryptPassword : " + cryptPassword);
    unCrypt = UserFunc.uDESDecrypt("", cryptPassword);
    UserFunc.uErrMsg(1,"UNCRYPT: " + unCrypt);
    // Using an explicitly given DES3 key
    cryptPassword2 = uDESEncrypt("qwertyuiopasdfghjklksdfr", "password"); /* Key
must be exactly 24 bytes. */
    UserFunc.uErrMsg(1,"cryptPassword2 : " + cryptPassword2);
    unCrypt2 = UserFunc.uDESDecrypt("qwertyuiopasdfghjklksdfr", cryptPassword2);
    UserFunc.uErrMsg(1,"UNCRYPT2: " + unCrypt2);
}
```

Related Information

[uEncrypt \[page 1149\]](#)

9.1.18.18 uEncrypt

Description:	Encrypts a string using the default encryption algorithm.	
Syntax:	OutString=uEncrypt(String ClearText[, String Algorithm[, String Key[, String CharacterEncoding]]]);	
Parameters:	ClearText	String to encrypt.
	Algorithm	Optional. Which encryption algorithm to use. You can specify one of the following values: CRYPT DES3 DES3CBC AES128/CBC AES192/CBC AES256/CBC If no encryption algorithm is specified, the default encryption algorithm is used.
	Key	Optional. Key to use when encrypting the string. If missing or empty string, the default key in keys.ini is used. In this case the key number used for encryption is returned as part of OutString.

CharacterEncoding Optional. Specifies which character set to use when converting the decoded bytes to the returned string, and must be the name of a supported character set.

Which character sets are supported varies from platform to platform, but every implementation of the Java platform is required to support the following standard character sets:

- US-ASCII
- ISO-8859-1
- UTF-8
- UTF-16BE
- UTF-16LE
- UTF-16

Other character sets may be available, but this requires a manual installation.

If this parameter is not specified, the default character set for the platform is used.

OutString Encrypted string. For details about the syntax of the encrypted string, see *SAP Identity Management Security Guide*.

Example

In this example we use the default encryption algorithm:

```
String ClearText;
String Cipher;
String Decrypted;

ClearText = "Text to be encrypted";

Cipher    = uEncrypt(ClearText);
Decrypted = uDecrypt(Cipher);
' --- Decrypted = "Text to be encrypted"
```

In this example we use a custom algorithm and key:

```
String ClearText;
String Cipher;
String Decrypted;
String Key;
String Algorithm;

ClearText = "Text to be encrypted";
Key       = "qwertyuiopasdfghjkljsdfr"; /* Key must be exactly 24 bytes. */
Algorithm = "DES3CBC";

Cipher    = uEncrypt(ClearText, Algorithm, Key);
Decrypted = uDecrypt(Cipher, Key);
' --- Decrypted = "Text to be encrypted"
```

Related Information

[Encryption \[page 68\]](#)

[SAP Identity Management Security Guide](#)

9.1.18.19 uErrMsg

Description:

ⓘ Note

It is recommended to use the functions [uError](#), [uInfo](#) and [uWarning](#).

Writes a message to the error log. This function allows the programmer to write error messages to the standard log/screen.

This message is also written to the event log, if specified in the job configuration.

Syntax: uErrMsg(long iType, String Msg);

Parameters:

iType	Error type.
	• 0: Info • 1: Warning • 2: Error

Msg	A message string.

Example

```
uErrMsg(2, "No problem");
```

9.1.18.20 uError

Description: Writes an error message to the log. Substitutes [uErrMsg](#).

To stop the job in case of an error, use [uSkip](#).

ⓘ Note

If tracing is enabled, any messages written by this function will also be included in the trace.

Syntax: uError(String Msg);

Parameters: Msg A message string.

Example

```
uError("My error message");
```

9.1.18.21 uExpandString

Description: Expands occurrences of %\$constant% and %attribute% in a string.

Syntax: OutString=uExpandString(String InString, DSEEntry Par[, Boolean StripMissing])

Parameters: InString The string to be expanded.

Par Optional. A DSEEntry. If Par is missing, only constants are expanded.

StripMissing Optional. Default False. True: Remove substitution strings for which no substitutions were found.

False: Do not remove substitution strings.

OutString The expanded string.

Example

```
function SendMail2(Par)
{
// This message contains the job constants %$EMAIL-WEB-HOMEPAGE%, %$EMAIL-
// ADMINISTRATOR%, the system value %$ddm.date%,
// and the entry attributes %$DISPLAYNAME% and %$MSKEYVALUE%
// They will be replaced with the corresponding values by the uExpandString
// function
var msg = "<html>\n<head>\n<meta http-equiv=\"Content-Language\" content=\"no-
bok\">\n" +
"<meta http-equiv=\"Content-Type\" content=\"text/html;
charset=windows-1252\">\n" +
"<title>Identity Center</title>\n</head>\n<body>\n<h1>Identity Center</h1>\n" +
"<p>Welcome %$DISPLAYNAME%. </p>\n<p>You are now a registered user of the
Identity Center, and can log in at\n" +
"<a href=\"%$EMAIL-WEB-HOMEPAGE%\">%$EMAIL-WEB-HOMEPAGE%</a>. Your login
identifier is %$MSKEYVALUE%. </p>\n<p>&nbsp;</p>\n" +
"<p><b><font size=\"4\">Best regards</font></b></p>\n<p>The MIC development
team</p>\n<p>&nbsp;</p>\n" +
"<p>PS. If you have any questions regarding this e-mail, please contact\n" +
"<a href=\"mailto:%$EMAIL-ADMINISTRATOR%\">%$EMAIL-ADMINISTRATOR%</a>.</p>\n" +
"<hr>\n<p><font face=\"Courier New\">ExampleOrg%$ddm.date%<br>\n" +
"<a href=\"http://www.example.com\">www.example.com</a></font></p>\n" +
"<p>&nbsp;</p>\n<p>&nbsp;</p>\n</body>\n</html>";
/* Expand occurrences of %$CONSTANT% and %$ATTRIBUTE% in string, using data from
Par. %XX% will be stripped if XX is not an attribute in Par */
```

```

var message = UserFunc.uExpandString(msg, Par, true);
var recipient = Par.get("mail");
recipient = recipient + ".nospam";
// remove above line to use real-email addresses
UserFunc.uSendSMTPMessage("%$EMAIL-ADMINISTRATOR%", recipient, "%$EMAIL-
SUBJECT%", message, "%$EMAIL-SMTP%", 1);
}

```

9.1.18.22 uFingerPrint

Description: Computes a fingerprint for the given entry, using the supplied delta info.

Syntax: FingerPrint=uFingerprint(DSEEntry Entry, String DeltaInfo);

Parameters:	Entry	The DSEEntry for which to create a fingerprint.
	DeltaInfo	The names of all attributes used to generate the fingerprint in the delta database. Can be retrieved from the delta database using the function uGetDeltaInfo .
	FingerPrint	The resulting fingerprint.

Example

```

// Main function: createfingerprint
// This is an entry script, computing the fingerprint, and storing
// it as a new attribute in the entry. The mInfoFields variable is
// initialized in the pass's Initialize script

function createfingerprint(Par)
{
var fp = uFingerprint(Par, mInfoFields);
UserFunc.uErrMsg(4, fp); /* Log it if loglevel is set to 4 or higher */
Par.put("fingerprint", fp); /* Compute fingerprint for the given entry, using
the supplied delta info */
return Par;
}

```

9.1.18.23 uFromBase64

Description: Converts a Base64 encoded string to normal text.

Syntax: OutString=uFromBase64(String Base64String[, String CharacterEncoding]);

Parameters: Base64String Base64 encoded string to be decoded.

CharacterEncoding	Optional. Specifies which character set to use when decoding the string, and must be the name of a supported character set. Which character sets are supported varies from platform to platform, but every implementation of the Java platform is required to support the following standard character sets: • US-ASCII • ISO-8859-1 • UTF-8 • UTF-16BE • UTF-16LE • UTF-16 Other character sets may be available, but this requires a manual installation. If this parameter is not specified, the default character set for the platform is used.
OutString	Returned string.

Example

```
Base64String = "ABCDEF"
OutString = uFromBase64(Base64String)
' --- OutString = <decoded string>
```

```
Base64String = "ABCDEF"
OutString = uFromBase64(Base64String, "ISO-8859-1")
' --- OutString = <decoded string>
```

9.1.18.24 uFromFile

Description:	Reads data from a file.						
Syntax:	ReadString=uFromFile(String FileName, long Length, String bBinary);						
Parameters:	<table> <tr> <td>FileName</td><td>Name of file to read from.</td></tr> <tr> <td>Length</td><td>Length to return. -1 means the whole file.</td></tr> <tr> <td>bBinary</td><td> True: Binary file False: Text file </td></tr> </table>	FileName	Name of file to read from.	Length	Length to return. -1 means the whole file.	bBinary	True: Binary file False: Text file
FileName	Name of file to read from.						
Length	Length to return. -1 means the whole file.						
bBinary	True: Binary file False: Text file						
ReadString	Returned data from file.						

Example

```
FileName = "c:\test.txt"
Length = -1
bBinary = FALSE
ReadString = uFromFile(FileName,Length,bBinary)
' --- ReadString = <contents of test.txt>
```

9.1.18.25 uFromHex

Description: Decodes a hex encoded string. A hex string contains twice the number of bytes of the decoded string.

Syntax: OutString=uFromHex(String HexString[, String CharacterEncoding]);

Parameters: HexString Hex string to be decoded.

CharacterEncoding Optional. Specifies which character set to use when converting the decoded bytes to the returned string, and must be the name of a supported character set.

Which character sets are supported varies from platform to platform, but every implementation of the Java platform is required to support the following standard character sets:

- US-ASCII
- ISO-8859-1
- UTF-8
- UTF-16BE
- UTF-16LE
- UTF-16

Other character sets may be available, but this requires a manual installation.

If this parameter is not specified, the default character set for the platform is used.

OutString Returned string.

Example

```
HexString = "ABCDEF"
OutString = uFromHex(HexString)
' --- OutString = <decoded string>
```

9.1.18.26 uGeneratePassword

Description:	Generates a random password.	
Syntax:	Password=uGeneratePassword(Int Length [,Boolean Pronounceable]);	
Parameters:	Length	The number of characters in the password.
	Pronounceable	Optional. If Pronounceable is false or missing, the password must contain upper and lowercase letters and digits.
	Password	The generated password.

Example

```
Password = UserFunc.uGeneratePassword(8);/* Generate an 8 character password. */
```

9.1.18.27 uGenHash

Description:

Generates a hashed string.

Note

If the value you are hashing is written by the Identity Management User Interface, it is converted to uppercase before it is saved. Make sure that the same attributes are handled consistently both when written by a job or from the Identity Management User Interface.

Syntax:

OutString=uGenHash(String Value[, String Algorithm[, String CharacterEncoding]]);

Parameters:

Value

String to be hashed.

Algorithm

Optional. Algorithm to use. You can choose between the following values:

MD5

SSHA

SHA256

SHA384

SHA512

PBKDF2_SHA1

If no algorithm is specified, the default hashing algorithm will be used.

CharacterEncoding

Note

Only available with the Java runtime engine.

Optional. Specifies which character set to use when converting the decoded bytes to the returned string, and must be the name of a supported character set.

Which character sets are supported varies from platform to platform, but every implementation of the Java platform is required to support the following standard character sets:

- US-ASCII
- ISO-8859-1
- UTF-8
- UTF-16BE
- UTF-16LE
- UTF-16

Other character sets may be available, but this requires a manual installation.

If this parameter is not specified, the default character set for the platform is used.

OutString

The hashed string. For details about the syntax of the hashed string, see *SAP Identity Management Security Guide*.

Example

In this example, we use the default hashing algorithm:

```
String ClearText;
String HashedValue;

ClearText = "MyPassword";

HashedValue=uGenHash(ClearText);
if uCompareHash(ClearText, HashedValue)
    /* Compare OK */
```

In this example we specify which hashing algorithm to use:

```
String ClearText;
String HashedValue;
String Algorithm;

ClearText = "MyPassword";
Algorithm = "SSHA";

HashedValue=uGenHash(ClearText, Algorithm);
if uCompareHash(ClearText, HashedValue)
    /* Compare OK */
```

9.1.18.28 uGetApprovers

Description: Returns a dictionary/Map of possible approvers for a given task and entry ID.

Syntax: `Approvers=uGetApprovers(Int MSKey, Int Task, String AttributeName [,Int AdminMSKEY]);`

Parameters:	MSKey	MSKey (Entry ID) of the entry that is to be approved.
	Task	Task ID of the approval task.
	AttributeName	The attribute to include, for example e-mail address or mobile phone number.
	AdminMSKEY	This value is used if the SQL statement containing %ADMINMSKEY%. Then %ADMINMSKEY% will be replaced by the value of this parameter.
	Approvers	A dictionary/Map of possible approvers, including the value of the AttributeName attribute.

Note

The function is only available within the provisioning module of the Identity Center.

Example

Java:

```
function GetApprovers(Par){
    mskey = UserFunc.uGetEntryID();
    approvers = uGetApprovers(mskey, 92, "USER_MAIL"); /* Returns the approvers
(MSKEY + DesiredAttribute) for a given entry and task */
    uErrMsg(1,approvers.size().toString());
    if (approvers.size() > 0)
    {
        mskeys = approvers.keySet().toArray();
        max = mskeys.length;
        for (i = 0; i < max; i ++){
            approvermskey = mskeys[i];
            uErrMsg(1,approvermskey);
            approvermail = approvers.get(approvermskey);
            UserFunc.uErrMsg(1,"Got approver mskey " + approvermskey + " with
email " + approvermail);
            UserFunc.uSendSMTPMessage("mymail@example.com", approvermail, "Task
ready for approval", "Task for entry ID: " + mskey + " ready for approval");
        }
    }
    return Par;
}
```

9.1.18.29 uGetAttrsOfType

Description:	Returns identity store attributes of the specified type.	
Syntax:	OutString=uGetAttrsOfType(Int IDStore, Int AttributeType);	
Parameters:	IDStore	The ID of the identity store.
	AttributeType	The ID of the attribute type.
	OutString	The names of the attributes separated by .

Example

```
Dim Result
Result=uGetAttrsOfType(1, 4)

Result will contain a pipe separated list of attributenames which are of type 4
(member-of attribute type).
Type 1 is General attribute
```

9.1.18.30 uGetAuditID

Description:	Returns the current audit ID. Only relevant when running Identity Center/Provisioning.	
Syntax:	Java: AuditID=uGetAuditID()	
Parameters:	AuditID	The current audit ID.

Example

```
AuditRef==ugetAuditID();

Auditref will contain audit reference number for current running provisioning
job.
```

9.1.18.31 uGetChangeValues

Description: Returns the old and new value when an attribute is changed. The function will normally be used in a provisioning task initiated by events on an attribute.

Syntax: OutString=uGetChangeValues([String Delimiter] [, String AuditID]);

Parameters: **Delimiter** Delimiter between the new and the old value. The default value is !.

AuditID Optional. Reference to an audit ID to the task that performed the change. The default value is 0 (current).

OutString <Attribute name>:<OPERATION>;<New value>!!<Old value>

Example

```
OutString=uGetChangeValues();  
  
LOCATION:INSERT;Oslo!!  
LOCATION:MODIFY;London!!Oslo  
LOCATION:DELETE;!!London
```

9.1.18.32 uGetConstant

Description: Returns the value of a constant (package constant, system constant or repository constant).

Syntax: Value=uGetConstant(String ConstantName);

Parameters: **ConstantName** The name of the constant (without % and \$).

<constantname> Job constant

ddm.<constantname> System constant

pck.<constantname> Package constant

glb.<constantname> Global constant

Note

When upgrading from SAP Identity Management 7.2 to 8.0, global constants are replaced by package constants. Nevertheless, constants can still be referred to with **%\$glb.% prefix** due to backward compatibility. Scripts that reference package constants with **%\$glb.% prefix** will continue to work, but you will get a warning in the Job Log.

rep.<constantname> Current repository constant

Value	The value of the constant.

Example

This is useful (or necessary) when a constant contains backslashes. For example:

The value of the constant \$FILENAME is "c:\newdata.txt". If we in a script write:

```
var Buffer = UserFunc.uFromFile("%$FILENAME%", 100, 0);
```

the characters \n will be interpreted as newline, because they are inside quotes. As a result, the file will not be found. If we write:

```
var Buffer = UserFunc.uFromFile(UserFunc.uGetConstant("FILENAME"), 100, 0);
```

this problem will not occur, because the \n will now be in a String, not in a string constant.

9.1.18.33 uGetContextVar

Description: Returns the value of a given context variable. See also [uAppContextVar](#), [uIncContextVar](#) and [uSetContextVar](#).

To get all context variables, use the function [uMGetContextVar](#).

Syntax: OutString=uGetContextVar(String Key[, String Default]);

Parameters:	Key	The name of the context variable.
	Default	Optional. A default value that is returned if the context variable is missing.
	OutString	The value of the context variable. If a default value is specified, and the context variable is missing, the default value is returned. Else an error message prefixed with !ERROR.

Example

```
uSetContextVar("MyContextVar", "ThisValue");
OutString = uGetContextVar(MyContextVar);
'OutString = ThisValue
```

Related Information

[Context Variables \[page 62\]](#)

[uAppContextVar \[page 1139\]](#)
[uIncContextVar \[page 1174\]](#)
[uSetContextVar \[page 1221\]](#)
[uMGetContextVar \[page 1199\]](#)

9.1.18.34 uGetCount

Description: Returns the number of entries processed in the current pass. Can be used in combination with function `uErrMsg` to give special info messages to the log file for each 1000 entries etc.

Syntax: `iCount=uGetCount()`

Parameters: `iCount` Number of processed entries.

Example

```
iCount=uGetCount()  
' --- iCount = <number of processed entries>
```

9.1.18.35 uGetDeltaInfo

Description: Returns the values of OwnerID and DeltaFields for a given delta identifier

Syntax: `DeltaInfo=uGetDeltaInfo(String DeltaIdentifier [,String DB]);`

Parameters: `DeltaIdentifier` The delta identifier for which information should be returned.

`DB` Optional. Connection string to the delta database. The default is the same as the Identity Center database.

`DeltaInfo` The owner ID and the names of all attributes written to the destination.

Example

```
// Main function: InitFingerprint  
// This is a pass INIT-script, where the fields used to create the fingerprint  
// MD5 hash  
// is retrieved from the DB, and stored in a global variable  
  
var mInfoFields;  
function InitFingerprint(Par)
```



```
{
url = uGetConstant("ddm.ddmpath") + "/fromldapdelta.mdb";
mInfoFields = uGetDeltaInfo("fromldap", url); /* Returns Info about the fields
used to create the delta fingerprint */
uErrMsg(0, mInfoFields); // log it.
}
```

9.1.18.36 uGetEntryID

Description:	Returns the ID of the current entry in the identity store. Only relevant when running Identity Center/Provisioning.	
Syntax:	CurrentID=uGetEntryID()	
Parameters:	CurrentID	ID of the current entry.

Example

```
CurrentID=uGetEntryID()
' --- CurrentID = <ID of current entry>
```

9.1.18.37 uGetErrCount

Description:	Gets the total number of errors up to this point. Can be used in combination with uStop to terminate a job if more than, for instance, 100 errors occur.	
Syntax:	iCount=uGetErrCount()	
Parameters:	iCount	Returned number of errors.

Example

```
iCount=uGetErrCount()
' --- iCount = <number of serious errors>
```

9.1.18.38 uGetErrorInfo

Description: Returns info about the last error that occurred. This function is intended to be called from the error handler script.

Syntax: errInfo =uGetErrorInfo();

Parameters: **errInfo** as DSEEntry object Information about the last error, or null if no error has occurred. The error info is returned in a DSEEntry object. The name of the DSEEntry returned will be a string describing the error (normally the same string that will be logged). See below for details.

Example

```
// This is an example of an ERRORHANDLER script calling uGetErrorInfo
function ERRORHANDLER(Par)
{
  thisErr = uGetErrorInfo();
  uErrMsg(3, "Got error - " + thisErr.getName());
  attr = thisErr.firstAttr();
  // In this example, we'll just dump all attributes to the log
  while (attr != null)
  {
    attrName = attr.getKey();
    attrValue = attr.getValue();
    uErrMsg(0, attrName + "=" + attrValue);
    attr = thisErr.nextAttr();
  }
}
```

Details

This object will always contain the following attributes:

Attribute	Value
err_category	"ERROR" or "WARNING".
err_where	A string description of where the error occurred.
err_classname	The Java class where the error occurred.
err_passname	The name of the pass where the error occurred, or the string "unavailable".

Different error situations may add other attributes like:

Attribute	Value
err_exception	A Java exception.
err_AddException	An exception from an Add operation.
err_ModException	An exception from a Mod operation.

Attribute	Value
err_Delexception	An exception from a Delete operation.
err_SQLWarning	An SQLWarning exception.

If the entry that caused the error was available when the error was detected, the entry will be merged with the error information.

9.1.18.39 uGetGeneralizedTimeZ

Description: Returns the system time relative to GMT in a standardized format that makes it possible to sort on date/time, for instance.

Syntax: ZTime=uGetGeneralizedTimeZ()

Parameters: ZTime as Variant System time in the format YYYYMMDDhhmmssZ, for instance 20011016083348Z

Example

```
ZTime = uGetGeneralizedTimeZ()
' --- ZTime = <time>
```

9.1.18.40 uGetGlobalVar

Description: Returns the value of a package variable.

Note

In previous releases, the function gets a global variable which is now represented as a package variable in version 8.0.

Syntax: OutString=uGetGlobalVar(String Key, String Default);

Parameters:	Key	The name of the package variable.
	Default	Default value if the specified key is not found.
	OutString	The value of the specified package variable.

Example

```
OutStr=uGetGlobalVar("GlobalVar","If no value")
```

9.1.18.41 uGetGUID

Description: Returns a Globally Unique Identifier looking like this:
"C2E51A4E-4318-4733-AEED-ACEBD7B90B90".

Syntax: OutString=uGetGUID()

Parameters: OutString A unique string (GUID).

Example

```
OutString=uGetGUID()  
' OutString="626630DB-F9F8-06D1-AB3A-4B3E8CF05AFE"
```

9.1.18.42 uGetIDStore

Description: Returns the ID of the current identity store if it is called from a provisioning job. Only relevant when running Identity Center/Provisioning.

Syntax: IDStoreID=uGetIDStore()

Parameters: IDStoreID ID of the current identity store if called from a provisioning job. Else 0.

Example

```
IDStoreID=uGetIDStore()
```

9.1.18.43 uGetIniFile

Description: Returns the value of an element in an .INI file.

Syntax: OutString=uGetIniFile(String iFile, String Section, String Key, String Default[, String CharacterEncoding]);

Parameters:	iFile	Name of .INI file, including directory.
	Section	Name of section in .INI file.
	Key	Name of specification in .INI file.
	Default	Default value if no specification matching the previous values is found.
	CharacterEncoding	Optional. Specifies which character set to use when converting the decoded bytes to the returned string, and must be the name of a supported character set. Which character sets are supported varies from platform to platform, but every implementation of the Java platform is required to support the following standard character sets: • US-ASCII • ISO-8859-1 • UTF-8 • UTF-16BE • UTF-16LE • UTF-16 Other character sets may be available, but this requires a manual installation. If this parameter is not specified, the default character set for the platform is used.
	OutString	The value of the specification (or the default value).

Example

```
OutStr=uGetIniFile("C:\temp\myini.ini", "Mysection", "MyKey", "If no value")
```

9.1.18.44 uGetPackageTaskID

Description: Returns the ID of the task based on PackageID and TaskName, or 0 if not found */");

Syntax: Value = uGetPackageTaskID (String packageId, String taskName)

Parameters:	PackageID	Id of the package. It must be numeric value.
	TaskName	Qualified name of the task

Example

```
Value = uGetPackageTaskID (NumericPackageId, "Task.Qualified.Name");
```

9.1.18.45 uGetPassDocumentation

Description:	Returns the description (documentation) of a pass.
Syntax:	OutString=uGetPassDocumentation()
Parameters:	OutString The documentation of the current pass. g

Example

```
OutString=uGetPassDocumentation()  
' --- OutString = <the documentation of the current pass>
```

9.1.18.46 uGetPassSubject

Description:	Gets the name of the current pass.
Syntax:	OutString=uGetPassSubject()
Parameters:	OutString The name of the pass.

Example

```
OutString=uGetPassSubject()  
' --- OutString = <the name of the current pass>
```

9.1.18.47 uGetProvisionUser

Description: Returns the UserID info from an entry in the provisioning audit log.

Syntax: OutString=uGetProvisionUser(Int AuditID);

Parameters: **AuditID** Audit identifier. In a provisioning task you can use uGetAuditID() to retrieve the current audit ID.

OutString Contents of the UserID column in the provisioning audit log. This field contains information about how the actual task was started.

Example

```
MyResult=uGetProvisionUser(uGetAuditID())
```

9.1.18.48 uGetRepositoryID

Description: Returns the ID of the current repository. Only relevant when running Identity Center/Provisioning.

Syntax: RepositoryID=uGetRepositoryID()

Parameters: **Repository ID** Repository ID of the current repository. If 0, repository is not set.

Example

```
RepositoryID=uGetRepositoryID()  
' --- RepositoryID = <Repository ID of the current repository>
```

9.1.18.49 uGetRepositoryName

Description: Returns the name of the current/specified repository. Only relevant when running Identity Management/Provisioning.

Syntax: OutString=uGetRepositoryName([String ReplD]);

Parameters:	RepID	The repository ID you want to return the name of. If not given, the name of the current repository is returned. Default empty.
	OutString	Name of the repository. Null if repository is not set.

Example

```
OutString=uGetRepositoryName()
' --- OutString = <Repository name of the current repository>
```

9.1.18.50 uGetRepositoryPackageID

Description:	Returns the ID of the package that repository type of current repository belongs to, or 0 if not applicable. No arguments are expected
Syntax:	Value = uGetRepositoryPackageID();
Parameters:	

Example

```
Value = uGetRepositoryPackageID()
```

9.1.18.51 uGetRepositoryVar

Description:	Returns the value of a repository variable. The repository is given by the job or task.				
Syntax:	OutString=uGetRepositoryVar(String Key);				
Parameters:	<table> <tr> <td>Key</td><td>The name of the repository variable.</td></tr> <tr> <td>OutString</td><td>The value of the specified repository variable.</td></tr> </table>	Key	The name of the repository variable.	OutString	The value of the specified repository variable.
Key	The name of the repository variable.				
OutString	The value of the specified repository variable.				

Example

```
OutString=uGetRepositoryVar("RepVar")
```

9.1.18.52 uGetTaskID

Description: Returns the task ID of the current task.
It does not return the task ID of an obsolete task.

Syntax: TaskID=uGetTaskID()

Parameters: TaskID Task ID of the current task.

Example

```
TaskID=uGetTaskID()  
' --- TaskID = <Task ID of the current task>
```

9.1.18.53 uGetTaskName

Description: Returns the task name for a given task.

Syntax: OutString=uGetTaskName([Int TaskID]);

Parameters: TaskID Optional. The task ID of a given task. If the parameter is 0 or empty, the task name of the current task is returned.

OutString The task name of the given task.

Example

```
OutString=uGetTaskName(5)  
' --- TaskID = <Task name of task 5>
```

9.1.18.54 uGetTaskRoot

Description: Returns the ID of the task that initiated the current provisioning (root task).

Syntax: OutString=uGetTaskRoot();

Parameters: OutString The task ID of the task that initiated the provisioning.
g

Example

```
OutString=uGetTaskRoot();  
' --- OutString = <Task ID of root task>
```

9.1.18.55 uGetUserVar

Description: Returns the value of a job variable.

Syntax: OutString=uGetUserVar(String Key, String Default);

<i>Parameters:</i>	Key	The name of the job variable.
	Default	Default value if the specified key is not found.
	OutString	The value of the specified job variable.

Example

```
OutStr=uGetUserVar("UserVar","If no value")
```

9.1.18.56 uGetWarnCount

Description: Gets the total number of warnings up to this point. (See also uGetErrCount.)

Syntax: iCount=uGetWarnCount()

Parameters: iCount Returned number of warnings.

Example

```
iCount=uGetErrCount()  
' --- iCount = <number of serious errors>
```

9.1.18.57 uHexToByte

Description: Decodes a hex value and returns byte[].

Syntax: OutString=uHexToByte(String HexString);

Parameters:	HexString	A string of hex digits in string form, optionally prefixed with {HEX}.
	g	
	OutString	The decoded string.
	g	

Example

```
OutString = uHexToByte("{HEX}48656c6c66");  
/* OutString will now contain the string "Hello" */
```

9.1.18.58 uHTMLEscape

Description: Replaces UNICODE characters in a string with the corresponding HTML escape characters.

Syntax: OutString=uHTMLEscape(String InString);

Parameters:	InString	The string to process.
	OutString	The processed string.
	g	

Example

```
OutString=UserFunc.uHTMLEscape("Måløy");  
' OutString="M&aring;l&oslash;y"
```

9.1.18.59 uHTMLUnescape

Description: Replaces HTML escape characters with UNICODE characters.

Syntax: OutString=uHTMLUnescape(String InString);

Parameters: InString The string to process.

OutString The processed string.

Example

```
OutString=UserFunc.uHTMLUnescape("M&aring;l&oslash;y")
'OutString="Måløy"
```

9.1.18.60 uIncContextVar

Description: Increments a context variable by 1 in one atomic operation. If the variable has no value, or the existing value is not numeric, 0 is assumed. Returns the new value. See also [uAppContextVar](#), [uGetContextVar](#) and [uSetContextVar](#).

Syntax: OutString=uIncContextVar(String Key);

Parameters: Key The name of the context variable.

OutString The new value of the context variable.

Example

```
int attemptNo = uIncContextVar("NumAttempts");
if (attemptNo == 10( {
    uError("Too many attempts");
}
```

9.1.18.61 uInfo

Description: Writes an info message to the DSE.log text file. Substitutes [uErrMsg](#).

The Info message is written in the DSE.log text file. The path to this file is the value of the [Work area](#) parameter in the [Job Log Detail](#) view. The Info message is not available in the HTML, XML and TEXT formats of the job logs that are displayed in the Identity Management Developer Studio and the Identity Management Administration User Interface.

To be able to see the Info messages in the job logs, on the [General](#) tab of the dispatcher, set the [Log Level](#) to [Info](#). This is also relevant for other log levels, such as [Trace](#) and [Debug](#).

Note

If tracing is enabled, any messages written by this function will also be included in the trace.

Syntax: uInfo(String Msg);

Parameters: Msg A message string.

Example

```
uInfo("My info message");
```

9.1.18.62 uInt8ToDate

Description: Converts data from Integer8 to a date format. This can be used to convert an Integer8 value from an attribute in Active Directory that is used to contain a date, for instance pwdlastset.

Syntax: OutString=uInt8ToDate(Number InString[, Boolean AdjustTimeZone]);

Parameters:

InString	The string to be converted.
AdjustTimeZone	Optional. Specifies that the date should be adjusted for the time zone.
one	True=Local time
	False= UTC
	Default is false.

OutString The string converted to a date in string form.

Example

```
function Convert2Date(par)
{
    return uInt8ToDate(Par,true);
}
```

9.1.18.63 ulS_Approval

Description: Processes an approval.

Syntax: OutString=ulS_Approval(Int Operation, String AdditionalParams, Int ApprovalID, Int Approver[, String Reason][, String ValidFrom][, String ValidTo]);

Parameters:	Operation	Type of operation: 1. Approve 2. Decline 3. Delegate 4. Add approver 5. Remove approver 6. Force escalation 7. Force decline 8. Authenticated approve 9. Authenticated decline 10. Authenticated delegate 11. Authenticated forced escalation 12. Authenticated forced decline

Note

Operation type 4 and 5 depends on the setting for "Allow change of appr" on the approval task.

Note

When using operation types 8 - 10, the authentication needs to be assured through the process prior to executing the function.

AdditionalParams	The value of this parameter depends on the specified Operation:

- Operation=3
The MSKEY of the user to whom the approval is delegated.
- Operation=4
The MSKEY of the user to be added as approver.
- Operation=5
The MSKEY of the approver to be removed.

ApprovalID	The ID of the approval. You can find this in the views idmv_approvals_basic/idmv_approvals_ext.

Approver The MSKEY of the user on behalf of whom the operation is performed. This user will be shown in the approval history.

For Operation type 2, 4 and 5 you can use 0 to specify <System> as processor.

Note

For Operation type 4 and 5, legal MSKEYs are defined by the value for "Allow change of appr" on the approval task.

Reason Optional. Reason for the operation.

ValidFrom Optional.

Note

Only used if Operation=1 (approval).

The Valid From date for the assignment in the form YYYY-MM-DD.

ValidTo Optional.

Note

Only used if Operation=1 (approval).

The Valid To date for the assignment in the form YYYY-MM-DD.

OutString Empty string or error message prefixed with !ERROR:.

Note

When using basic approvals, note the following:

- Operations 4 (Add approver) and 5 (Remove approver) are not allowed.
- Operation 6 (Force escalation) does not require approver.
- Operations 1 (Approve), 2 (Decline), 3 (Delegate), 7 (Force decline), 8 (Authenticated approve), 9 (Authenticated decline), 10 (Authenticated delegate), 11 (Authenticated forced escalation) and 12 (Authenticated forced decline) require approvers calculation. The calculation itself is done when the approvers activate the *To Do* tab in the Identity Management User Interface. Alternatively, you can call the stored procedure `mxp_check_approvals` with input parameter the approver's mskey. This stored procedure calculates the approvals for the current input mskey. For more information on how to execute a stored procedure using SAP Identity Management Developer Studio, see SAP Note:

[2974026](#)

Example

```
// Main function: DeclineApproval
function DeclineApproval(Par){
    var UniqueId= Par.get("ApprovalId");
    var Result = uIS_Approval(2,"",UniqueId,0,"Declined by system");
    if (Result.indexOf("!ERROR") >= 0) {
        uError(Result);
    }
}
```

```

    }
    return;
}

```

In this example the function is used to decline (Operation=2) the approval with a given ApprovalID (UniqueID). The ApprovalID is found in the view idmv_approvals_ext.

The processor is set to 0 (<System>) and the decline reason (Declined by system).

The "SQL statement" on the "Source" tab for this pass finds all approvals for a given user (MSKEYVALUE):

```
select distinct(ApprovalId) from idmv_approvals_ext
```

```
where approvalType=1
```

```
and ApproverMSKEYVALUE='<MSKEYVALUE>'
```

Related Information

[Approval Processing \[page 34\]](#)

9.1.18.64 uIS_Compare

Description: Attributes in the identity store can be defined as "Compare only". uIS_Compare can be used to compare all types of attributes to a specified value.

Syntax: OutString=uIS_Compare(Int MSKey, Int IStore, String AttName, String AttValue);

Parameters:	MSKey	The entry's ID (MSKey).
	ISStore	The ID of the identity store.
	AttName	The name of the attribute to compare.
	AttValue	The value to compare to.
	OutString	True (<>0), False (0). If the requested attribute does not exist, False is returned.

Example

```
OutString=uIS_Compare(25, 1, "telephone", "3456746")
```


9.1.18.65 ulS_Create

Description: Creates an entry in the identity store.

ⓘ Note

It is strongly recommended to use a To identity store pass for updating attributes in the identity store. For performance reasons and to avoid deadlocks in the database, no entry event tasks are started, and it is not verified that the attribute is unique. Updating attributes in this way may lead to duplicates in the identity store.

If you use a script to calculate an attribute value, the script could be used to return the result to the attribute definition in the To identity store pass instead.

Syntax: Java: OutString=ulS_Create(Int IStore, String EntryValue);

Parameters:	ISStore	The ID of the identity store.
	EntryValue	The value for the MSKeyValue attribute. This must be unique.
	OutString	The MSKey of the created entry or error message prefixed with !ERROR:

Example

```
OutString=uIS_Create(1, "EntryValue")
```

9.1.18.66 ulS_Delete

Description: Deletes an entry from the identity store. The attribute will be kept in the history store.

Syntax: OutString= ulS_Delete(Int IStore, String EntryValue);

Parameters:	ISStore	The ID of the identity store.
	EntryValue	The value for the MSKeyValue attribute for the entry that is to be deleted.
	OutString	The MSKey of the deleted entry, or error message prefixed with !ERROR:.

Example

```
OutString=uIS_Delete(1, "78459")
```

9.1.18.67 uIS_Get

Description: Returns the MSKEY of an entry from an identity store.

Syntax: OutString=uIS_Get(Int iStore, String EntryValue);

Parameters: iStore The ID of the identity store.

EntryValue The value for the MSKEYVALUE attribute for the entry that is to be returned. Case insensitive.

OutString The MSKey of the returned entry, or error message prefixed with !ERROR:..

Example

```
OutString=uIS_Get(1, "Administrator")
'OutString = 250
```

Returns the MSKEY of the user Administrator.

9.1.18.68 uIS_GetEntry

Description: Returns all attributes from an entry (MSKEY) as dictionary. Multivalue attributes are separated by |.

Syntax: Result=uIS_GetEntry(Int MsKey[, String Delimiter);

Parameters: MsKey The entry's ID (MSKey).

Delimiter Optional. The delimiter that is to be used as separator in multivalue attributes. Default value is |.
The Java version only supports |.

Result Java: A hash map (DSEEntry) containing all attributes from an entry.

ⓘ Note

The usage of this function could impact performance. In case only several attributes are needed, use [uIS_nGetValues \[page 1183\]](#) instead.

ⓘ Note

This function is only available with the Identity Center.

Example

Sample Code

```
Function getEntry(par)
Set result=uIS_getEntry(123)
for each MyItem in result
MyVar=MyItem & ";" & result(MyItem)
'.....
```

9.1.18.69 uIS_GetError

Description: Returns the error description according to the error number.

Syntax: ErrString=uIS_GetError(Int ErrID);

Parameters:	ErrID	The error number.
	ErrString	The error description corresponding to the error ID.

Example

```
ErrString=uIS_GetError(2)
```

9.1.18.70 uIS_GetValue

Description: Returns the value of a given attribute for a given entry in the identity store.

Syntax: OutString=uIS_GetValue(Int MSKey, Int IStore, String AttName);

Parameters:	MSKey	The entry's ID (MSKey).
	ISore	The ID of the identity store. If this is 0, the correct identity store is found using the MSKey.
	AttName	The name of the attribute.
	OutString	The value of the attribute or error message prefixed with !ERROR:.

Example

```
OutString=uIS_GetValue(25, 0, "MyAttr")
```

9.1.18.71 uIS_LinkEntries

Description:	When reading information from directories like Active Directory, it is possible to link user entries to groups and groups to groups to create the same hierarchy in the identity store as in the source directory. This function is intended to be used with the template "Link entries" found in the folder \Identity Center\Jobs\Link entries.		
Syntax:	OutString=uIS_LinkEntries(Int iStore, String LinkAttribute, String ChildAttribute, String ChildValue, String ParentAttribute, String ParentValue[, Int Remove] [, Int OwnerRepository];		
Parameters:	iStore	The ID to the identity store.	
	LinkAttribute	The attribute that is used to link the entries.	
	ChildAttribute	The attribute in the sub-entry that is used when linking it to the parent.	
	ChildValue	The value of the ChildAttribute that is to be matched with the value in ParentValue.	
	ParentAttribute	The attribute in the parent entry that is used when linking it to the sub-entries.	
	ParentValue	The value of the ParentAttribute that is to be matched with the value in ChildValue.	
	Remove	Optional. Indicates whether to try to create a link if an error occurs. 0=Try to create a link 1=Delete link Default is 0.	
	OutString	The value of the attribute or error message prefixed with !ERROR:.	

Example

Example (3 passes):

1. Read groups from AD into identity store. Use the sub-table function for the member attribute to keep the link information.
2. Read persons into identity store
3. Connect the entries using the link information in the sub-table

Group entry:

```
mskeyvalue=mygroupname
mx_entrytype=group
orgdn=cn=mygroup,cn=users,dc=myorg,dc=com
description=any group description
```

User entry:

```
mskeyvalue=myuser
mx_entrytype=person
givenname=ole
sn=Olsen
userdn=cn=ole_olsen,cn=users,dc=myorg,dc=com
```

Use the link template to connect these entries:

```
result=uis_linkentries(1,"mxref_group","userdn","cn=ole
olsen,cn=users,dc=myorg,dc=com","orgdn","cn=mygroup,cn=users,dc=myorg,dc=com",0)
```

If the result starts with an "ERROR:" it is an error message. The last parameter 0 indicates that the link reference is set. 1 indicates to remove any link between the entries.

The user entry will look like this:

User entry:

```
mskeyvalue=myuser
mx_entrytype=person
givenname=ole
sn=Olsen
userdn=cn=ole_olsen,cn=users,dc=myorg,dc=com
mxref_group=123
```

where 123 is the unique MSKEY for group entry.

9.1.18.72 uIS_nGetValues

Description: Returns the values of a given attribute for a given entry in the identity store.

Syntax: OutString=uIS_nGetValues(Int MSKey, String AttName[, String Separator]);

Parameters:	MSKey	The entry's ID (MSKey).
	AttName	The name of the attribute.
	Separator	Optional. Separator for multivalue attributes. The default value is "!!".
	OutString	The values of the attribute, empty if not present. In case of any exception, error message prefixed with ERROR:.

Example

```
OutString=uIS_nGetValues(25, "MyAttr");
```

Here is an example of how you can split the result in separate values:

```
// Main function: logEmails
function logEmails(Par){
    // Retrieve any additional email addresses, separated by !!
    // Par is assumed to be the MSKEY of the entry.
    addresses = uIS_nGetValues(Par, "MX_MAIL_ADDITIONAL", "!!");
    if (addresses.indexOf("!!ERROR")>= 0)
    {
        uError(addresses); // Log the error message
    }
    else
    {
        // Split the returned string on the same delimiter as specified
        vector = uSplitString(addresses , "!!");
        if (vector != null && !vector.isEmpty())
        {
            for (i = vector .iterator(); i.hasNext();)
            {
                value = i.next();
                uInfo(value); // Log the email address
            }
        }
    }
    return Par;
}
```

9.1.18.73 uIS_PrivReconcile

Description:

Normally, when you are changing roles and privileges that affect assignments, the affected entries are marked "dirty" and will be reconciled by a scheduled procedure.

In the cases where such changes are not automatically marked "dirty", the reconciliation must be initiated manually. You can also initiate a reconciliation manually to make sure that some (or all) entries are reconciled.

In the "Source" tab of the pass, you can for instance select all MX_PERSON entries, or all entries matching some other criteria that indicates that the entries should be reconciled.

Syntax: OutString=uIS_PrivReconcile(String EntryID[, Int Option]);

Parameters: EntryID The entry's ID (MSKey).

Option 0: Reconcile only failed assignments.

1: Reconcile failed and declined assignments.

The default value is 0

OutString Returns the status of the reconciliation or error message prefixed with !ERROR:
The status is returned in the form:
Entry: <mskey> Privilege add: <n> Privilege delete: <n>
where the number of adds is the number of "missing" privileges added to the entry and the number of deleted privileges is the number of privileges removed from the entry as a result of the reconciliation.

Example

In a provisioning job:

```
Result=uIS_PrivReconcile(uGetEntryID())
```

In a standard job:

```
Result=uIS_PrivReconcile(Par)
```

where Par is MSKey

9.1.18.74 uIS_RepairEntry

Description: Repairs a given entry in the identity store by reconciling and rebuilding all assignments for this entry. The process will check against the internal data structure and rebuild assignments for the entry based on that information. It will add missing assignments, retry failed assignments and calculate validity for inherited assignments.

This is a more costly process than only reconciling the entry, but can be necessary if a reconcile is not sufficient to correct. After the function completes for an entry, this entry is repaired. If this results in new assignments, any event tasks for these assignments will be started. If the job is processing many entries, the whole process may take a considerable amount of time and resources on the system.

Syntax:	OutString = uIS_RepairEntry(int MSKEY);				
Parameters:	<table><tr><td>MSKEY</td><td>The entry's ID (MSKey).</td></tr><tr><td>OutString</td><td>Empty string or error message prefixed with !ERROR:.</td></tr></table>	MSKEY	The entry's ID (MSKey).	OutString	Empty string or error message prefixed with !ERROR:.
MSKEY	The entry's ID (MSKey).				
OutString	Empty string or error message prefixed with !ERROR:.				

Example

```
p = 19905;  
OutString = uIS_RepairEntry(p);  
uInfo("Result: " + OutString);
```

9.1.18.75 uIS_ResolveDynamicGroup

Description: Resolves a dynamic group. After the function is executed, all entries which match the filter will be linked to the dynamic group.

Syntax: OutString=uIS_ResolveDynamicGroup(String EntryID[, boolean Test);

Parameters:	EntryID	The MSKey of the dynamic group to resolve.
	Test	Optional. When this flag (true/false) is set, the function will only return the result of performing the resolve, without actually resolving anything.
	OutString	Textual description or error message prefixed with !ERROR:. The returned text is: Group resolved OK: Members: <NumMembers> Added: <NumAdded> Deleted: <NumDeleted> Errors: <NumErrors>

Example

```
GroupID=245
OutString = uIS_ResolveDynamicGroup(GroupID);
'Outstring= Group resolved OK: Members: 54 Added: 2 Deleted: 0 Errors: 0
```

The function resolves the dynamic group identified with the MSKEY 245. During the process, two members were added. The total number of members after the resolve is 54.

9.1.18.76 uIS_Revert

Description: Sets an attribute or entry in the identity store to a given time/date.

Syntax: OutString=uIS_Revert(Int MSKey, String AttrName, String Date[, String UserID]);

Parameters:	MSKey	The entry's ID (MSKey).
	AttrName	The name of the attribute. One specific attribute can be reverted to the given time when attribute name is given. If attribute name is empty, the whole entry is reverted to the state given by the date.

Date	The date to revert to. The date must be specified in the following format: YYYY-MM-DD YYYY-MM-DD HH:MM:SS (24-hour clock, i.e. HH is 00...23) Examples of legal dates: 2007-02-15 2007-02-15 11:02:14 2006-12-24 17:30:00
UserID	Optional. General User ID or message that will be inserted in the field USERID in the audit log. Typically this can be distinguished name. The default value is empty.
OutString	Empty string or error message prefixed with !ERROR:.

Example

Revert "title" attribute:

```
Myresult=uIS_revert(123,"title", "2006-11-17 12:26:34.450","Revert by job #5")
```

Revert entry:

```
Myresult=uIS_revert(123,"", "2006-11-17 12:26:34.450","Revert by job #5")
```

9.1.18.77 uIS_SetApproval

Description:	Creates an approval for a given entry. For details about the MX_APPROVALS attribute, see the document <i>Identity Store Schema</i> .
Syntax:	OutString=uIS_SetApproval(Int EntryID, String Task, Int Audit, Int Approver, Int Operation[, String Reason]);
Parameters:	EntryID MSKey of the approvee.
	Task Task ID or Task GUID (preferable when using export/import) of the approval task.
	Audit Audit ID of the request.
	Approver MSKey of the approver.

Operation	Operation code. 0: Wait 1: Approved 2: Declined
Reason	Optional. Reason for decline.
OutString	Empty string or error message prefixed with !ERROR:.

Example

```
ApproveeMSKey=45
Task="001EAEFA-000F-496B-B8BA-540C7E8506CF"
AuditID=77
ApproverMSKey=93
Operation=1
OutString = uIS_SetApproval(ApproveeMSKey, Task, AuditID, ApproverMSKey,
Operation);
'OutString = ""
```

The approval task given with the GUID in "Task" for the entry given with ApproveeMSKey and approver given with ApproverMSKey is executed with Operation 1, meaning that the request is approved.

Related Information

[Identity Store Schema](#)

9.1.18.78 uIS_SetDirty

Description: Sets a given entry "dirty" to be picked up by the scheduled procedure for reconciliation.

This mechanism can be used for the entry types MX_PERSON, MX_PRIVILEGE, MX_ROLE, MX_GROUP, MX_DYNAMIC_GROUP.

If an MX_PERSON entry is set dirty, this entry is marked dirty and all assignments will be reconciled.

If an entry of one of the other (container) entry types is marked dirty, its member entries are marked dirty and will be reconciled.

Normally entries are automatically set "dirty" when structural changes are done to roles and privileges, but in some cases it is necessary to set this explicitly, for instance if the changes are done to the repository.

Syntax:	OutString = uIS_SetDirty(int MSKEY);	
Parameters:	MSKEY	The entry's ID (MSKey).
	OutString	Empty string or error message prefixed with !ERROR:.

Example

```
p = 19905;
OutString = uIS_SetDirty(p);
uInfo("Result: " + OutString);
```

9.1.18.79 uIS_SetValue

Description: Updates the value of a given attribute for a given entry in the identity store.

Note

Due to unpredictable results with using this function, it is strongly recommended to use a To identity store pass for updating attributes in the identity store. For performance reasons and to avoid deadlocks in the database, no entry event tasks are started, and it is not verified that the attribute is unique. Updating attributes in this way may lead to duplicates in the identity store.

Note

When using uIS_SetValue to update the value of a given attribute, provisioning will be triggered only if you update another attribute that is defined on the *Destination* tab of the same To Identity Store pass, from where a script using this function is called. This applies regardless of whether the given attribute is trigger attribute or not.

For example, if you use uIS_SetValue to update the value of the MX_LASTNAME attribute of a user, to provision your changes, you need to update the value of another attribute, for example MX_DEPARTMENT, that must be defined on the *Destination* tab of the To Identity Store pass.

If you use a script to calculate an attribute value, the script could be used to return the result to the attribute definition in the To identity store pass instead.

Syntax:	OutString=uIS_SetValue(Int MSKey, Int IStore, String AttName, String AttValue);	
Parameters:	MSKey	The entry's ID (MSKey).
	ISore	The ID of the identity store.
	AttName	The name of the attribute to be modified.

AttValue	The new value of the attribute. If the parameter is empty, the attribute is deleted and copied to historical data. If the parameter is NULLATTR, changes to the attribute are ignored. If the attribute is defined to be either hashed or encrypted, the value must be hashed or encrypted before it is stored with the functions uGenHash or uEncrypt.
UserID	Optional. General user ID or message that will be inserted in the field USERID in the audit log. Typically this can be a distinguished name. The default value is empty.
AttrOperation	Optional. Specifies which operation to perform. Possible values are: 0: Multivalue: Add Single value: Replace or add (if not existing) 1: delete case sensitive 2: delete case insensitive The default value is 0.
OutString	If an event task is defined on the attribute in the identity store, the function returns the Audit ID in the provisioning audit log. If no event task is defined on the attribute, the function returns an empty string. Else error message prefixed with !ERROR:.

Example

```
OutString=uIS_SetValue(21, 1, "MyAttr", "MyValue")
```

9.1.18.80 uIS_sGet

Description: Returns an entry from the identity store based on the value of a given attribute.

Syntax: OutString=uIS_Get(Int iStore, String Attribute, String AttValue[, Boolean ExactMatch]);

Parameters: **iStore** The ID of the identity store.

Attribute The name of the attribute to use as a criterion.

AttValue The value to search for. Can include wild characters, for instance "Ron%".

ExactMatch Optional. Specifies whether the search is performed with LIKE or =. Possible values:

False: Search using "Like". The search will not differentiate between _ and -. (Default)

True: Exact match (=).

OutString The MSKey of the returned entry, number of occurrences (negative number) or error message prefixed with !ERROR:.

Example

```
OutString=uIS_sGet(1, "MX_LASTNAME", "br%");
```

9.1.18.81 uIS_sGetValue

Description: Returns the current value of an attribute of a given entry in an identity store or the previous value of the same attribute.

Syntax: OutString=uIS_sGetValue(Int MSKey, Int IStore, String AttName, Int Level);

Parameters:	MSKey	The entry's ID (MSKey).
	ISore	The ID of the identity store. If this is 0, the correct identity store is found using the MSKey.
	AttName	The name of the attribute.
	Level	Attribute version. Can have one of the following values: 0: Current value 1: Previous value
	OutString	The value of the attribute or error message prefixed with !ERROR:..

Example

```
mail=uIS_sGetValue(123, 0, "mail", 0)  
' Returns the current value of the attribute.
```

9.1.18.82 uIS_Touch

Description: (Re)calculates the expiry time for an attribute in the identity store.

Syntax: OutString=uIS_Touch(Int MSKey, Int IStore, String AttName, String AttValue, Int TouchType [,Int TimeToLive]);

Parameters:	MSKey	The entry's ID (MSKey).
	ISore	The ID of the identity store. If this is 0, the correct identity store is found using the MSKey.
	AttName	The name of the attribute.

AttValue	The attribute value.
TouchType	The algorithm for calculating the expiry time. Possible values are: 0: Expiry time is set according to the attribute's TTL setting 1: Expiry time is set to now (timeout) 2: Expiry time is set to forever (no timeout) 3: Expiry time is set to now + the value of the TimeToLive parameter
TimeToLive	Optional. Used if TouchType is set to 3.
OutString	Empty string, or error message prefixed with !ERROR:.

9.1.18.83 uIsRunning

Description: Detects whether the runtime engine is running a full job or inserting a data source template.

Syntax: Value=uIsRunning()

Parameters: Value as
Long
0: Not running
1: Running

Example

```
if uIsRunning() {
    'Running a complete job
    ...
} else {
    'Doing insert template
    ...
}
```

9.1.18.84 uKillFiles

Description: Deletes files.

Syntax: uKillFiles(String DeleteFiles);

Parameters: DeleteFiles All files matching the specified pattern will be deleted.

Note

In the Java version, it is only possible to specify one specific file. It does not support wildcards.

Note

There is no UNDO for this function. The files are permanently deleted. A parameter like this "C:*.*" is fatal!

Example

```
uKillFiles("c:\job\*.log");
```

This will delete all files with the extension .log in the directory c:\job\.

9.1.18.85 uLDAPGetDN

Java runtime engine

Description: Returns the DN returned by the specified LDAP URL if only one entry matches the URL. The function can be used in any pass type.

Syntax: OutString=uLDAPGetDN(String URL [,String User] [,String Password] [,boolean FollowReferrals]);

Parameters:	URL	LDAP URL.
	User	Optional. Valid user name for the directory server.
	Password	Optional. Password for the specified user name.
	FollowReferrals	Optional. True if you wish the LDAP call to follow referrals. The default is NOT to follow referrals.
	OutString	String containing the DN. Else: !ERROR: #0 if no entries matched the URL !ERROR: #<n> where <n> is the number of matching entries !ERROR: EXCEPTION: <java exception message>

Example

```
OutString = uLDAPGetDN("LDAP://directory.example.com/o=example.com??SUB?
(guid=4829450)")
```

9.1.18.86 uLDAPGetEntry

Description: Returns the entry returned by the specified LDAP URL if one entry matches the URL. The function can be used in any pass type.

Syntax: Entry=uLDAPGetEntry(String URL [,String User] [,String Password] [,boolean FollowReferrals]);

Parameters:	URL	LDAP URL.
	User	Optional. Valid user name for the directory server.
	Password	Optional. Password for the specified user name.
	FollowReferrals	Optional. True if you wish the LDAP call to follow referrals. The default is NOT to follow referrals.
	Entry	DSEEntry containing the entry, or null if not found. If there is more than one entry, only one entry will be returned. The attribute "err_category" contains the value "WARNING"; the attribute "err_where" contains a text message stating how many entries were returned. If an exception occurs, the entry will have the attribute err_category set to "ERROR", and the attribute err_exception will contain the exception.

Note

In the attributes part of the LDAP URL (as specified in RFC 2255) you can force the attribute to be retrieved as binary by prefixing the attribute name with an exclamation mark (!).

Example

```
var entry = uLDAPGetEntry("ldap://directory.example.com /cn=Lisa
Andersson ,O=Myorg,C=com??BASE?(objectclass=*)"); /* Get the DSEEntry returned
by given URL */
if (entry != null)
{
    category = 0;
    if (entry.containsKey("err_category"))
    {
        category = 2;
        if (entry.get("err_category").equals("WARNING"))
        {
            category = 1;
        }
    }
}
```



```

}
uErrMsg(category, "---Dumping entry data--");
attr = entry.firstAttr();
// In this example, we'll just dump all attributes to the log
while (attr != null)
{
    attrName = attr.getKey();
    attrValue = attr.getValue();
    uErrMsg(category, attrName + "=" + attrValue);
    attr = entry.nextAttr();
}
}

```

9.1.18.87 uLocaleToSapLanguage

Description: Converts an SAP Locale string code to an SAP two-letter language code. See also [uSapLanguageToLocale](#).

Syntax: OutString=uLocaleToSapLanguage(String Locale);

Parameters: Locale Locale string.

OutString Two-letter language code.

Example

```

OutString = uLocaleToSapLanguage("zh_CN");
'OutString = ZH

```

Mapping of uLocaleToSapLanguage

See the mapping that is performed in the coding in the table below:

Code	Result
IW	HE
IN	ID
zh_TW	ZF
zh_HK	ZF
Anything else (for example, en_US)	The same is returned, in uppercase (for example, EN)

Related Information

[uSapLanguageToLocale \[page 1212\]](#)

9.1.18.88 uLookupClose

Description: Closes the active lookup database.

In a job, uLookupOpen can be defined in the INITIALIZE function and uLookupClose in the TERMINATE function.

Syntax: uLookupClose();

Parameters:

9.1.18.89 uLookupCreateDB

Description: Creates a new lookup database.

Syntax: uLookupCreateDB(String dbName, String iTable, String iIndex, iName As Variant, iKey As Variant, iNumber As Variant [,Special As Boolean]);

Parameters: **dbName** Java: Correct JDBC URL to the database.

iTable Table name.

iIndex Name of primary index.

iName Key field.

iNumber Name of number field.

Special Optional. Determines whether to create a lookup database for uLookupSeek or uLookupSSeek.
True: uLookupSSeek
False: uLookupSeek

Example

```
Function initialize(Par)
' Create Lookup Database (if not exists)
' and Open Lookup DB
call uLookupCreateDB("%$ddm.ddmpath%\lookup.mdb",
"lookuptb", "dbIndex", "Name", "Key", "Number", TRUE)
Res=uLookupOpen("%$ddm.ddmpath%\lookup.mdb", "lookuptb")
```

```
initialize = ""  
End Function
```

9.1.18.90 uLookupDelete

Description: Deletes an entry in the lookup database.

Syntax: OutString=uLookupDelete(String pLookIn, String pLookFor);

Parameters:	pLookIn	Field to search.
	pLookFor	Record to search for.
	OutString	Empty if the operation succeeded, otherwise the error code is returned.

Example

```
Outstring=uLookupDelete("dbIndex","keyvalue")
```

9.1.18.91 uLookupOpen

Description: Opens a Microsoft Access database for lookup. Returns 0 if the operation succeeded, otherwise an error code is returned.

Note

Only one lookup database can be active at a time.

In a job uLookupOpen can be defined in the INITIALIZE function and [uLookupClose](#) in the TERMINATE function.

Syntax: iErr=uLookupOpen(String DatabaseName, String TableName);

Parameters: **DatabaseName** Java: Correct JDBC URL to the file name.

TableName Name of database table to open.

iErr Status returned.

Example

```
DatabaseName = "%$ddm.ddmpath%\db.mdb"
TableName = "tbName"
iErr= uLookupOpen(DatabaseName,TableName)
' --- iErr = <status>
```

9.1.18.92 uLookupSeek

Description: This lookup function can be used as an alternative to sub-queries. The lookup database must be a Microsoft Access database opened by uLookupOpen. Returns the lookup key associated with the in parameter. Returns an empty string if there is no mapping.

The function is used where there is a unique string in the source data that should be mapped to a unique number or other unique key. This may be an e-mail address that is mapped to a unique number. The lookup database must exist, but may be empty.

Syntax: iMapKey=uLookupSeek(String LookIn, String LookFor, String LookRet, String LookNew, Boolean MakeNew);

Parameters:	LookIn	Index name of the column for where to search.
	LookFor	Look for string.
	LookRet	Return column.
	LookNew	If MakeNew is True, a new entry is added if no matching entries are found. LookNew contains the value you want to insert in the return column when creating new entries. If LookRet is an Autonumber column, a new value is generated automatically. LookNew is then set to "".
	MakeNew	Makes a new entry in the Lookup database if no matching entry is found. True: Make new entry False: Do not make new entry
	iMapKey	The value associated with the string given in LookFor.

Example

Given a lookup table with two columns, an indexed key and an Autonumber. The first column is called EMAIL, which is a text field containing 100 characters. The other column is UID.

```
LookIn = "EMAIL"
LookFor = cstr(par)
LookRet = "UID"
LookNew = ""
MakeNew = TRUE

iMapKey = uLookupSeek (LookIn,LookFor,LookRet,LookNew,MakeNew)
' --- iMapKey = <the contents of UID for the e-mail address given in PAR>
```

9.1.18.93 uMD5

Note

As of *SAP NetWeaver Identity Management Identity Center 7.2*, it is recommended to use the function [uGenHash](#) instead.

Description: MD5-encodes a string. The string result will always be a unique 32-byte hex-encoded string.	
Syntax: OutString=uMD5(String InString[, String CharacterEncoding]);	
Parameters:	InString String to parse
	CharacterEncoding Optional. Specifies which character set to use when converting the encoding the string, and must be the name of a supported character set. Which character sets are supported varies from platform to platform, but every implementation of the Java platform is required to support the following standard character sets: • US-ASCII • ISO-8859-1 • UTF-8 • UTF-16BE • UTF-16LE • UTF-16 Other character sets may be available, but this requires a manual installation. If this parameter is not specified, the default character set for the platform is used.
OutString Returned MD5 of 32 bytes.	

Example

```
InString = "Madame"
OutString = uMD5(InString)
' --- OutString = <hex-encoded string>
```

9.1.18.94 uMGetContextVar

Description: Returns all context variables matching the given filter as a Java properties map. See also uGetContextVar . By default, all context variables are retrieved, but this can be reduced by using the filter.	
Syntax: OutString=uMGetContextVar([String Filter]);	

Parameters:	Filter	Optional. If this contains a string, it will return all context variables starting with this string. If empty, all context variables are returned.
	OutString	Returned context variables as a Java properties map. Else an error message prefixed with ! ERROR.

Example

```
vars = uMGetContextVar();
varstring = vars.toString();
```

If the above is used for an Add member event for a privilege, the varstring will contain the following string:

```
{#MxLinkReference=15937, #MxEventAttrValue_1=203, #MxEventAttrCount=1,
#MxEventType=15, #MxEventAttrName_1=MXREF_MX_PRIVILEGE}
```

#MxEventType 15 means that this is an Add member task

203 is the MSKEY of the privilege being added, and 15937 is the unique ID in the link table.

Processing the properties map can be done in the following way:

```
// get all context variables where name starts with #MSG
ctxVars = uMGetContextVar("#MSG%");
var ctxkeyset = ctxVars.keySet ();
var keyname = "";
for (keys = ctxkeyset.iterator (); keys.hasNext ();dummy=1)
{
    keyname = keys.next ();
    keyvalue = ctxVars.getProperty (keyname);
    uInfo("Key:"+keyname+ " value:"+keyvalue);
}
```

9.1.18.95 uNewEntry

Description: When using a From Generic pass you need to create a DSEEntry that can be used to return the result of the pass.

Syntax: DSEEntry=uNewEntry(Name);

Parameters: **Name** Name of entry, i.e. distinguished name.

DSEEntry A DSEEntry that can be returned with the result of the pass.

Example

```
function GETNEXTENTRY(Par){
var result = null;
// The mDSML object is declared in the "Open data source" script
var entry = mDSML.getNextEntry()
if (entry != null)
{
    // create DSEEntry object to return to the script
    result = uNewEntry(entry.get("userid"));
    var attr = entry.nextAttr();
    while (attr != null)
    {
        // add one attribute to the returned object
        result.put(attr.getKey(), attr.getValue());
        attr = entry.nextAttr();
    }
}
return result;
}
```

9.1.18.96 uPreparedSelect

Description:	Returns the result of a Select statement in the Identity Management database.				
Syntax:	Result = uPreparedSelect(sqlStatement, typeValueArg1, typeValueArg2,...,typeValueArgN);				
Parameters:	<table><tr><td>Query</td><td>SQL statement: The SQL statement to be executed with question mark (?) as a placeholder for all values in the WHERE clause</td></tr><tr><td></td><td>SQL statement arguments: Ordered <java.sql.Type>/<value> pairs for each (?) placeholder in the provided SQL statement</td></tr></table>	Query	SQL statement: The SQL statement to be executed with question mark (?) as a placeholder for all values in the WHERE clause		SQL statement arguments: Ordered <java.sql.Type>/<value> pairs for each (?) placeholder in the provided SQL statement
Query	SQL statement: The SQL statement to be executed with question mark (?) as a placeholder for all values in the WHERE clause				
	SQL statement arguments: Ordered <java.sql.Type>/<value> pairs for each (?) placeholder in the provided SQL statement				
Result	The result of the execution of uPreparedSelect procedure is an ArrayList object containing the selected rows and columns. The returned ArrayList object is never NULL and you could check the number of returned rows by using the following field: result.length				

uPreparedSelect uses java.sql.PreparedStatement to execute the provided SQL statement. To take advantage of the java.sql.PreparedStatement, instead of providing values directly into the WHERE clause of the SQL statement, use the (?) placeholder. The values must be provided to uPreparedSelect as a separate list of type/value pairs in the following format:

```
<java.sql.Type>/<value>
```

The following java.sql.Type argument types are supported: 'VARCHAR', 'NVARCHAR', 'LONGVARCHAR', 'NCHAR', 'INTEGER', 'BOOLEAN', 'TIMESTAMP', 'DATE'. The provided argument values can be NULL for all argument types. They can be empty only for the 'VARCHAR', 'NVARCHAR', 'LONGVARCHAR', 'NCHAR' types.

Note

If you want to use `uPreparedSelect` for SQL statements with not supported argument types, you can still do it by providing the argument value directly into the SQL statement. But, in this way, you will not be able to take advantage of the `java.sql.PreparedStatement`.

Note

If you retrieve values from the identity store, it is recommended to use one of the views installed with the product.

To avoid performance problems due to SQL statements given with this function, the query is analyzed and you see a warning in the job log. Specifically note that `avalue` should not be used after "where" or "order by" as `avalue` is not an indexed column.

Avoid queries like:

```
select mskey from idmv_value_simple where avalue like 'Hello%'
select mskey from idmv_value_simple order by avalue
```

Instead use the indexed column `searchvalue` after "where" or "order by", where possible.

```
select avalue from idmv_value_simple where searchvalue like 'HELLO'
select mskey from idmv_value_simple order by searchvalue
```

Using the `searchvalue` column is also necessary to avoid problems with case sensitivity when using an Oracle database. The values in the `searchvalue` column is always converted to upper case.

Note

When using Microsoft SQL Server, it is recommended that all SELECT queries should include "with (nolock)". This will reduce the table locking, improve performance, and reduce the probability of deadlocks. For instance:

```
SELECT <columns> FROM <table> WITH (NOLOCK) WHERE <column>=<value>
```

Example

This is an example of javascript that uses `uPreparedSelect` :

```
// Main function: uPreparedSelectTestScript
function uPreparedSelectTestScript(Par) {
    var mskeyRange = 10;
    var mskeyvalueRange = "MX_PRIV:WD:%";
    var sqlStatement = "SELECT * FROM idmv_entry_simple WHERE mcMSKEY < ? AND mcMSKEYVALUE LIKE ? AND mcIdStore = ? AND mcDirty = 0";

    var result = UserFunc.uPreparedSelect(sqlStatement, "INTEGER/" + mskeyRange, "NVARCHAR/" + mskeyvalueRange, "INTEGER/1");
    if (result.length > 0) {
        // Means return the 7th column from the 4th row from the ResultSet (counting from 0)
        return result[3][6];
    }
}
```



```
}  
}
```

9.1.18.97 uProvision

Description: Initiates a provisioning task for a given entry in the identity store.

Syntax: `AuditRef=uProvision(Int MSKey, String Task, Int RefAudit, Int Repository, String UserID, Int Delay[, Int Standalone]);`

Parameters: **MSKey** The entry's ID (MSKey).

Note

The **MSKey** value must be greater than **0**.

Task	TaskID, TaskGUID, or Qualified Name of the task to be initiated.
RefAudit	Reference audit, if available. If the function is called from a provisioning job, the audit reference of this task can be submitted to the task initiated by uProvision and inserted in the audit log. Use 0 if no reference audit is available.
Repository	Repository ID. 0 means no repository.
UserID	General user ID or message that will be inserted in the field USERID in the audit log. Typically this can be a distinguished name or Active Directory login name.
Delay	Delay in seconds until the task should be initiated.
Standalone	Optional. Normally, when a task is started with uProvision and RefAudit is given, the task will be an event task of the original task. This parameter specifies whether the task should be an event task or not. Possible values: 0: The task is started as an event task. (Default) 1: The task is started "standalone" (not as an event task). For more information about how to use context variables when a task is started by uProvision, see Context Variables [page 62].
AuditRef	Audit reference or error message prefixed with !ERROR:.

Example

```
MyAudit=uProvision(2,5,0,0,"",50);  
MyAudit=uProvision(2,'5',0,0,"",50);  
MyAudit=uProvision(2,"5",0,0,"",50);
```

Run task 5 on MSKey 2 and wait for 50 seconds before the task is initiated.

→ Tip

When you initiate a provisioning task for a given entry and you are using **TaskID** for task specification, you may use a string enclosed in single quotes ('), double quotes ("), or a number.

Example

```
MyAudit=uProvision(1, "535ECB5B-04FE-4AAE-9D73-52DDE9ABC8", 0, 0, "", 30);
```

Run task with GUID **"535ECB5B-04FE-4AAE-9D73-52DDE9ABC8"** on MSKey 1 and wait for 30 seconds before the task is initiated.

Example

```
MyAudit=uProvision(3, 'com.sap.MyPackage:task.MyTask', 0, 0, "", 20);
```

Run task with qualified name **'com.sap.MyPackage:task.MyTask'** on MSKey 3 and wait for 20 seconds before the task is initiated.

→ Tip

When you initiate a provisioning task for a given entry, you may use **TaskGUID** and **Qualified Name** with single quotes (') or double quotes (").

9.1.18.98 uPutIniFile

Description: Updates an entry in an .INI file.

Syntax: uPutIniFile(String iFile, String Section, String Key, String Value);

Parameters: **iFile** Name of .INI file to be updated, including directory.

Section Section name in specified file.

Key Name of specification to be updated.

Value New value.

Example

```
uPutIniFile("c:\temp\myini.ini", "MySection", "MyKey", "Hello world");
```

9.1.18.99 uReFormatDate

Description:	Converts a date to a string using the given format. The last three parameters are optional.	
Syntax:	OutString=uReformatDate(String Date, String OutputFormat [,String InputFormat] [,String Language] [,String Country]);	
Parameters:	Date	Date in a string form.
	OutputFormat	Format to use for returned string. For a full description of the date and time patterns, see the java documentation for SimpleDateFormat.
	InputFormat	Optional. Format used in input string.
	Language	Optional. Language code (needed for some formats, such as names of weekdays).
	Country	Optional. Country code (needed for some formats).
	OutString	String in new form.

Example

```
Result = uReFormatDate("010203", "'EEE, d MMM yyyy", "'ddMMyy");
```

/* Result should now contain the string "Sat, 1 Feb 2003" */

9.1.18.100 uReplaceBinary

Description:	Replace function for binary string. uReplaceBinary requires an exact match (is case sensitive). The function uReplaceString is case insensitive.	
Syntax:	OutString=uReplaceBinary(String InString, String SearchFor, String ReplaceBy);	
Parameters:	InString	Original string.
	SearchFor	String to be searched for.
	ReplaceBy	Replace with string.
	OutString	New string with replacement.

Example

```
InString = "Madame"
```

```
SearchFor = "Ad"
ReplaceBy = "AD"
OutString = uReplaceBinary(InString, SearchFor, ReplaceBy)
' --- OutString = "Madame"
```

This function will return "Madame" and will not perform any operation on the string, because "Ad" does not match any part of "Madame" exactly.

```
InString = "Madame"
SearchFor = "ad"
ReplaceBy = "AD"
OutString = uReplaceBinary(InString, SearchFor, ReplaceBy)
' --- OutString = "MADame"
```

This function will return "MADame" because the string "ad" is found.

9.1.18.101 uReplaceString

Description: Replaces a string. This function does not require an exact match (it is case insensitive). See also [uReplaceBinary](#).

Syntax: OutString=uReplaceString(String InString, String SearchFor, String ReplaceBy);

Parameters:	InString	Original string.
	SearchFor	String to be searched for.
	ReplaceBy	Replace with string.
	OutString	New string with replacement.

Example

```
InString = "Madame"
SearchFor = "Ad"
ReplaceBy = "AD"
OutString = uReplaceString(InString, SearchFor, ReplaceBy)
' --- OutString = "MADame"
```

9.1.18.102 uRepositoryConfig

Description: Set the repository state. Returns empty string or error message prefixed with !ERROR:

Syntax: Value = uRepositoryConfig(String repositoryName, String disableFromDate, String disableToDate, String reasonString, <optional>String scheduleRuleName);

Parameters:	RepositoryNameString	Name of the repository
	DisableFromDate	Date Value in ISO 8601 format to set to disable from
	DisableToDate	Date Value in ISO 8601 format to set to disable to
	ReasonString	Reason for the operation

Example

```
Value = uRepositoryConfig("repositoryName", "2014-10-18", "2014-10-19",  
"Planned maintenance");
```

9.1.18.103 uRepositoryStatus

Description: Checks repository active state. Returns status as string or error message prefixed with !ERROR:

Syntax: value = uRepositoryStatusRe(String repositoryName);

Parameters: **RepositoryNameString** Name of the repository

Example

```
Value = uRepositoryStatusRe("repositoryName")
```

9.1.18.104 uResetDelta

Description: Resets the delta information for a given OwnerID in the delta database.

Syntax: uResetDelta(String OwnerID, String DeltaURL);

Parameters: **OwnerID** The OwnerID for which delta information should be reset.

DeltaURL The URL to the delta database.

Example

```
uResetDelta("ActiveDirectory", %$ddm.identitycenter%);
```

9.1.18.105 uResolveDGMembership

Description: Checks and resolves whether a given entry is a member of a dynamic group. After executing the function, the entry will be linked to the group if it matches the filter. The returned value indicates if there was any change in the status.

See also [uIS_ResolveDynamicGroup](#).

Syntax: OutString=uResolveDGMembership(Int Group, Int Child);

Parameters:

Group	The MSKey of the dynamic group.

Child	The MSKey of the child entry.

OutString The function returns one of the following values:

g	<0: Error

	0: Not member

	1: Member

	2: Not member, but was referenced before the check

	3: Member, but was not referenced before the check

Example

```
Group = 41  
Child = 151  
OutString = uResolveDGMembership(Group, Child);  
'OutString = 3
```

The entry identified with the MSKEY 151 was not a member of the dynamic group before, but was added during the resolve process.

9.1.18.106 uRetryPrivilegeAdd

Description: Retries a privilege assignment for a given entry. This is useful if the privilege assignment failed, for instance because the target application was unavailable at the time of the original privilege assignment. The view MXPV_Priv_AddStatus can be used to return all privilege assignments that have failed, and be used together with this function to retry all failed privilege assignments.

This can be a periodic job that retries all failed privileges. It is also possible to redo the privilege assignment for all users with a given privilege if the provisioning tasks have been changed.

See also [uRetryPrivilegeRemove](#).

Syntax:	OutString=uRetryPrivilegeAdd(Int MSKey, Int Privilege);	
Parameters:	MSKey	MSKey to the user.
	Privilege	MSKey to the privilege.
	OutString	Empty string or error message prefixed with !ERROR:.

Example

Create a job with the following configuration:

Source:

```
SELECT MSKEY,aValue FROM MXPV_Priv_AddStatus WHERE (ProvStatus = 1001 or
ProvStatus = 1101) and aValue=1234
```

This statement returns all failed privilege assignments for the privilege with MSKEY=1234

Destination:

Run a script which calls uRetryPrivilegeAdd(MSKEY,aValue)

All the failed privilege tasks will then be retried

9.1.18.107 uRetryPrivilegeRemove

Description: Retries a privilege removal for a given entry. This is useful if the privilege removal failed, for instance because the target application was unavailable at the time of the original privilege removal. The view MXPV_Priv_RemoveStatus can be used to return the privilege removal that has failed, and be used together with this function to retry the failed privilege removal.

This can be a periodic job that retries a failed privilege removal. See also [uRetryPrivilegeAdd](#).

Syntax:	OutString=uRetryPrivilegeRemove(uniqueLinkId);

Parameters: **uniqueLinkId** Unique identifier of the link between a user and a privilege.

OutString Empty string or error message prefixed with !ERROR:.

Example

Create a job with the following configuration:

Source:

```
SELECT uniqueLinkId FROM MXPV_Priv_RemoveStatus WHERE (ProvStatus = 1001 or  
ProvStatus = 1101) and MSKEY=5555 and aValue = 4444
```

This statement returns the uniqueLinkId of the failed privilege removal for user with MSKEY=5555 and privilege with MSKEY=4444.

Destination:

Run a script which calls uRetryPrivilegeRemove(uniqueLinkId)

This will run the deprovision task for the failed removal of privilege 4444 for user 5555.

9.1.18.108 uRunJobNow

Description: Sets an Identity Center job ready to run.

Syntax: OutString=uRunJobNow(JobID)

Parameters: **JobID** The ID of the job.

OutString An empty string (OK) or error message prefixed with !ERROR:.

Example

```
OutString=uRunJobNow(75)
```


9.1.18.109 uSapLang1ToLanguage

Description: Converts a SAP one-character language code to an SAP two-letter language code. See also [uSapLanguageToLang1](#).

Syntax: OutString=uSapLang1ToLanguage(String Lang1);

Parameters: Lang1 One-character SAP language code.

OutString Two-letter language code or null if Lang1 is not found.

Example

```
OutString = uSapLang1ToLanguage("D");  
'OutString = DE
```

To handle values not found:

```
OutString = uSapLang1ToLanguage("Daaa");  
if(OutString==null)  
{  
    uWarning("Mynull");  
}  
else  
{  
    uWarning(OutString);  
}
```

9.1.18.110 uSapLanguageToLang1

Description: Converts a two-letter SAP language code to a SAP one-character language code. See also [uSapLang1ToLanguage](#).

Syntax: OutString=uSapLanguageToLang1(String Lang);

Parameters: Lang Two-letter language code.

OutString One-character SAP language code or null if Lang is not found.

Example

```
OutString = uSapLanguageToLang1("DE");  
'OutString = D
```

```
OutString = uSapLanguageToLang1("UK");
'OutString = 8
```

9.1.18.111 uSapLanguageToLocale

Description: Converts a two-letter language code to a Locale string code. See also [uLocaleToSapLanguage](#).

Syntax: OutString=uSapLanguageToLocale(String Lang[, String Country]);

Parameters:	Lang	Two-letter language code.
	Country	Optional. Country code.
	OutString	Locale string code.

Example

```
OutString = uSapLanguageToLocale("ZH");
'OutString = zh_CN
```

Mapping of uSapLanguageToLocale

See the mapping that is performed in the coding depending on whether the country is specified or not:

- If the country is specified, then the result is `code_COUNTRY`, where code is always lowercase and COUNTRY is always uppercase. For example, `en_US`, `de_DE` and others. This is a standard `java.util.Locale` behavior.
- If the country is not specified, then the result is the following:

Code	Result
HE	iw
ID	in
ZF	zh_TW
ZH	zh_CN
Anything else (for example, EN)	The same is returned, in lowercase (for example, en)

Related Information

[uLocaleToSapLanguage](#) [page 1195]

9.1.18.112 uSelect

Description:	Returns the result of a Select statement in the Identity Management database.	
Syntax:	Result=uSelect(String Query);	
Parameters:	Query	The complete SQL query to be executed.
	Result	The result of the SQL query as a string. Values are delimited with a pipe character (). Entries are delimited with two exclamation marks.

Note

If you retrieve values from the identity store, it is recommended to use one of the views installed with the product.

To avoid performance problems due to SQL statements given with this function, the query is analyzed and you see a warning in the job log.

Specifically note that avalue should not be used after "where" or "order by" as avalue is not an indexed column.

Avoid queries like:

```
select mskey from idmv_value_simple where avalue like 'Hello%'
select mskey from idmv_value_simple order by avalue
```

Instead use the indexed column searchvalue after "where" or "order by", where possible.

```
select avalue from idmv_value_simple where searchvalue like 'HELLO'
select mskey from idmv_value_simple order by searchvalue
```

Using the searchvalue column is also necessary to avoid problems with case sensitivity when using an Oracle database. The values in the searchvalue column is always converted to upper case.

Note

When using Microsoft SQL Server, it is recommended that all SELECT queries should include "with (nolock)". This will reduce the table locking, improve performance, and reduce the probability of deadlocks. For instance:

```
SELECT <columns> FROM <table> WITH (NOLOCK) WHERE <column>=<value>
```

9.1.18.113 uSendMail

Description:	Sends an SMTP mail to given recipient(s). This function is implemented with JavaMail API . It provides the same functionality as uSendSMTPMessage function, which is obsolete. In addition, it supports sending a mail with multiple attachments.	
Syntax:	<pre>uSendMail(<Sender>, <Recipients>, <Subject>, <Message or FileName>, <SMTPHost>, [<ContentType>], [<Port no>], [<Attachments>]);</pre> <pre>ReturnCode=uSendMail(<Sender>, <Recipients>, <Subject>, <Message or FileName>, <SMTPHost>, [<Con- tentType>], [<Port no>], [<Attachments>]);</pre>	
Parameters:	Sender	E-mail address to set as originator. Can be specified as: Nickname <e-mail address>, for instance: Lisa Andersson <lisa.andersson@myorg.com>
	Recipients	Recipient addresses. Can be specified using the same syntax as for the sender. Delimit multiple recipients with a comma (,) or a semicolon (;). To specify a cc or bcc recipient, enter 'cc:' or 'bcc:' in front of the recipient's name. For instance: user1@mail.com; cc: user2@mail.com; bcc: user3@mail.com This value results as follows: To: user1@mail.com Cc: user2@mail.com Bcc: user3@mail.com Note If 'cc:' or 'bcc:' is not specified in front of the e-mail address, the recipient will be a part of the 'To' list.
	Subject	The subject of the message.
	MessageOrFileName	Message text or a fully qualified file name referencing a file to be used as a message.
	SMTPHost	Optional. Host of SMTP server. Default value: localhost

ContentType	Optional. Valid MIME content type. Accepts 0 and 1. You can also specify other content type value. 0 - text/plain 1 - text/html; charset=ISO-8859-1, or custom, for example: 'text/plain; charset=UTF-8', 'text/html; charset=UTF-8', ... A charset could be specified in the content type. For instance: 'text/plain; charset=UTF-8'. In case you specify fully qualified file name for MessageOrFileName parameter to be used as message, this charset is used to read the content of the file.
PortNo	Optional. Port to connect to SMTP server. Default value: 25
Attachments	Optional. Attachments file paths. Multiple attachments could be included in the mail. Delimit multiple attachments with a comma (,) or a semicolon (;).
ReturnCode	• 0 - Mail is sent successfully. • -1 - An error or exception has occurred during the attempt to send the mail.

Example

```
uSendMail('sender@mycompany.com', 'recipient@mycompany.com; cc:
recipient2@mycompany.com; bcc: recipient3@mycompany.com', 'Subject', 'Email
body', 'mailserver.mycompany.com');
```

```
uSendMail('sender@mycompany.com', 'recipient@mycompany.com', 'Subject', 'C:\
\Users\\Desktop\\DOCUMENTS\\EmailBody.txt', 'mailserver.mycompany.com');
```

```
uSendMail('sender@mycompany.com', 'recipient@mycompany.com', 'Subject', 'Email
body', 'mailserver.mycompany.com', '0', '25', 'C:\\Users\\Desktop\\DOCUMENTS\
attachment.zip, C:\\Users\\Desktop\\DOCUMENTS\\attachment.pdf');
```

9.1.18.114 uSendMailAuth

Description:	Sends an SMTP mail to given recipient(s) with authentication to SMTP server. This function is implemented with JavaMail API . It supports different configuration options and sends a mail with multiple attachments.		
Syntax:	<pre>uSendMailAuth(<Sender>, <Recipients>, <Subject>, <Message or FileName>, <Username>, <Password>, <Configuration>, <SMTPHost>, [<ContentType>], [<Port no>], [<Attachments>]);</pre> <pre>ReturnCode=uSendMailAuth(<Sender>, <Recipients>, <Subject>, <Message or FileName>, <Username>, <Password>, <Configuration>, <SMTPHost>, [<ContentType>], [<Port no>], [<Attachments>]);</pre>		
Parameters:	Sender	E-mail address to set as originator. Can be specified as: Nicename <e-mail address>, for instance: Lisa Andersson <lisa.andersson@myorg.com>	
	Recipients	Recipient addresses. Can be specified using the same syntax as for the sender. Delimit multiple recipients with a comma (,) or a semicolon (;). To specify a cc or bcc recipient, enter 'cc:' or 'bcc:' in front of the recipient's name, for instance: user1@mail.com; cc: user2@mail.com; bcc: user3@mail.com This value results as follows: To: user1@mail.com Cc: user2@mail.com Bcc: user3@mail.com	
 Note If 'cc:' or 'bcc:' is not specified in front of the e-mail address, the recipient will be a part of the 'To' list.			
	Subject	The subject of the message.	
	MessageOrFileName	Message text or a fully qualified file name referencing a file to be used as a message.	
	Username	Username used to authenticate to SMTP server.	
	Password	Password used to authenticate to SMTP server.	

Configuration

Defines the configuration (set of properties) used by **JavaMail API** to connect to SMTP server. There are three predefined configurations: SMTP, SMTPS, SMTP_TLS. Also, a custom set of properties could be specified.

If no value is specified, the default value is **<SMTP>**.

The predefined configurations have the following characteristics:

- SMTP
 - The connection to mail server is over SMTP.
 - STARTTLS upgrade is optional. It is executed only if mail server supports it.
 - The connection to mail server fallbacks to plaintext when the mail server does not indicate support for STARTTLS.
 - Certificate issuer checks are disabled.
 - Certificate identity checks are disabled.

JavaMail API properties used to connect to the mail sever:

```
mail.transport.protocol="smtp",
mail.smtp.starttls.enable="true",
mail.smtp.starttls.required="false",
mail.smtp.ssl.trust="*",
mail.smtp.ssl.checkserveridentity="false"
```

- SMTPS
 - The connection to mail server is over SMTPS. (SMTP entirely encapsulated by TLS).
 - Strict validation of server certificates is enabled.
 - Server certificates must be issued: by a certificate authority in the system trust store and to a subject matching the identity of the remote SMTP server.

JavaMail API properties used to connect to the mail sever:

```
mail.smtp.ssl.enable="true",
mail.smtp.ssl.checkserveridentity="true",
mail.smtp.quitwait="false"
```

- SMTP_TLS
 - The connection to mail server is over SMTP, but STARTTLS upgrade is mandatory.
 - The connection to mail server does not fallback to plaintext when the mail server does not indicate support for STARTTLS.
 - Strict validation of server certificates is enabled.
 - Server certificates must be issued: by a certificate authority in the system trust store and to a subject matching the identity of the remote SMTP server.

JavaMail API properties used to connect to the mail sever:

```
mail.transport.protocol="smtp",
mail.smtp.starttls.enable="true",
mail.smtp.starttls.required="true",
mail.smtp.ssl.checkserveridentity="true"
```

- **<CUSTOM_VALUE>**

If none of the above values is specified for Configuration parameter, the value is considered as **<CUSTOM_VALUE>**. With this custom value you could specify

your own set of properties to connect to mail server via **JavaMail API**. Multiple properties are delimited with a comma (,), for instance:

property1=value1, property2=value2, ...

For full list of available **JavaMail API** properties refer to the JavaMail API documentation.

Note

In the implementation of `uSendMailAuth`, several properties are always used to connect to SMTP server regardless of the specified configuration:

- `mail.smtp.host` - SMTPHost parameter value is set as value for this property.
- `mail.smtp.port` - PortNo parameter value is set as value for this property.
- `mail.smtp.auth` - It is set to "true", because username and password are used to authenticate to mail server.

SMTPHost	Optional. Host of SMTP server. Default: localhost
ContentType	Optional. Valid MIME content type. Accepts 0 and 1. You can also specify other content type value. 0 - text/plain 1 - text/html; charset=ISO-8859-1, or custom, for example: 'text/plain; charset=UTF-8', 'text/html; charset=UTF-8', ... A charset could be specified in the content type. For instance: 'text/plain; charset=UTF-8'. In case you specify fully qualified file name for MessageOrFileName parameter to be used as message, this charset is used to read the content of the file.
PortNo	Optional. Port to connect to SMTP server. Default: 25
Attachments	Optional. Attachments file paths. Multiple attachments could be included in the mail. Delimit multiple attachments with a comma (,) or a semicolon (;).

ReturnCode

- 0 - Mail is sent successfully.
- -1 - An error or exception has occurred during the attempt to send the mail.

Example

```
uSendMailAuth('sender@mycompany.com', 'recipient@mycompany.com', cc:
recipient2@mycompany.com, bcc: recipient3@mycompany.com', 'Subject',
'Email body', 'SmtMailServerUsername', 'password123!', 'SMTP',
'mailserver.mycompany.com');
```

```
uSendMailAuth('sender@mycompany.com', 'recipient@mycompany.com', 'Subject',
'C:\\Users\\Desktop\\DOCUMENTS\\EmailBody.txt', 'SmtMailServerUsername',
'password123!', 'SMTP', 'mailserver.mycompany.com');
```

```
uSendMailAuth('sender@mycompany.com', 'recipient@mycompany.com', 'Subject',
'<h1>Email body</h1>', 'SmtMailServerUsername', 'password123!',
'SMTPS', 'mailserver.mycompany.com', 'text/html; charset=UTF-8', '465',
'C:\\Users\\Desktop\\DOCUMENTS\\attachment.zip, C:\\Users\\Desktop\\DOCUMENTS\\
attachment.pdf');
```

9.1.18.115 uSendSMTPMessage

Description: Sends an SMTP message to the given recipient(s).

Note

This function is obsolete. We recommend using some of the new functions uSendMail and uSendMailAuth that provide extended functionality and authentication options. They are implemented with JavaMail API.

Syntax: uSendSMTPMessage(Sender, Recipients, Subject, MessageOrFileName, SMTPHost [,ContentType] [,PortNo] [,Attachment] [,AttachmentType]);

ReturnCode=uSendSMTPMessage(Sender, Recipients, Subject, MessageOrFileName, SMTPHost [,Content-Type] [,PortNo] [,Attachment] [,AttachmentType]);

Parameters:

Sender	E-mail address to set as originator.
	Can be specified as:
	Nicename <e-mail address>, for instance:
	Lisa Andersson <lisa.andersson@myorg.com>

Recipients

Recipient addresses. Delimit multiple recipients with a comma (,) or semicolon (;). To specify a cc or bcc recipient, enter cc: or bcc: in front of the recipient name, for instance "Rec1;cc:Rec2;bcc:Rec3".

In addition, when sending an e-mail to multiple recipients, enclose each e-mail address in brackets < >, for instance:

<user1@mail.com>;cc:<user2@mail.com>;cc:<user3@mail.com>

To: user1@mail.com

Cc: user2@mail.com, user3@mail.com

Note

If cc: or bcc: is not specified in front of the e-mail address, the recipient will be a part of the "To" list.

Can be specified using the same syntax as for the sender.

Subject	The subject of the message.
MessageOrFileName	Message text, or a fully qualified file name referencing a file to be used as the message.
SMTPHost	Localhost is default.
ContentType	Optional. Valid MIME content type. Accepts 0 or 1 as for VB. Default: 'text/plain;'
PortNo	Optional. Default: 25
Attachment	Optional. Fully qualified file name of a file to be included as an attachment.
AttachmentType	Optional. Valid MIME attachment type.
ReturnCode	0 - Message is sent successfully. -1 - An error or exception has occurred during the attempt to send the message.

Example

```
uSendSMTPMessage("sender@mycompany.com", "recipient@mycompany.com", "Test",  
"This message was sent from the Java runtime engine using SMTP",  
"mailserver.mycompany.com");
```

Related Information

[uSendMail \[page 1214\]](#)

[uSendMailAuth \[page 1216\]](#)

9.1.18.116 uSendSNMPTrap

Description: Sends the specified SNMP trap message to the SNMP manager specified in the Job options.

Syntax: uSendSNMPTrap(Int ErrType, String Text);

Alternate syntax:

uSendSNMPTrap(Int ErrType, String Text, String EnterpriseOID, Int GenericType, Int SpecificType, String MessageOID [, String Host] [, Int Port] [, String Community] [, Int TimeTicks];

Parameters:	ErrType	A number that can have the following values:
	e	3: Error
		2: Warning
		1: Info

	Text	A descriptive text.

Example

```
uSendSNMPTrap(1, "Passed test");
```

When using the short syntax, the following defaults are used:

Text will be prefixed with "DSE:"

EnterpriseOID will be 1.3.6.1.4.1.9331

GenericType will be 6 (Enterprise specific),

SpecificType will be 5

MessageOID will be 1.3.6.1.4.1.9331.4.<ErrType>

Host will be localhost

Port will be 162

Community will be "public"

TimeTicks will be time since the runtime engine started in hundredths of seconds

9.1.18.117 uSetContextVar

Description: Updates the value of a context variable. Any existing value is overwritten. If the context variable does not exist, it is created. See also [uAppContextVar](#) and [uIncContextVar](#).

Syntax:	uSetContextVar(String Key, String Value);	
Parameters:	Key	The name of the context variable.
	Value	The new value.

Example

```
uSetContextVar("MyContextVar", "MyValue");
```

9.1.18.118 uSetExitCode

Description:	Sets an error level before the runtime engine exits.	
Syntax:	uSetExitCode(Int ExitCode);	
Parameters:	ExitCode	A numeric value used to indicate the error level.

9.1.18.119 uSetGlobalVar

Description: Updates the value of a package variable.

Note

In previous releases, the function sets a global variable which is now represented as a package variable in version 8.0.

Syntax:	uSetGlobalVar(String Key, String Value);	
Parameters:	Key	The name of the package variable to be updated.
	Value	The new value.

Example

```
uSetGlobalVar("MyVar", "NewText");
```

9.1.18.120 uSetJobToError

Description:	Sets the current running job to error state permanently.	
Syntax:	OutString=uSetJobToError(String msg);	
Parameters:	msg	If message provided this will be logged in system log.
	OutString	If error encountered a message starting with !ERROR: <error text> will be returned. If empty the function executed OK.

Example

```
// Main function: SetJobToError
function SetJobToError(Par){
    //Example calling DSE internal function
    Myreturn=UserFunc.uSetJobToError("Set in error state");
    // Verify result from Myreturn...
    return;
}
```

9.1.18.121 uSetRepositoryVar

Description:	Updates the value of a repository variable.	
Syntax:	uSetRepositoryVar(String Key, String Value);	
Parameters:	Key	The name of the repository variable to be updated.
	Value	The new value.

Example

```
uSetRepositoryVar("MyVar", "NewText");
```

9.1.18.122 uSetUserVar

Description: Updates the value of a job variable.

Syntax: uSetUserVar(String Key, String Value);

Parameters:	Key	The name of the job variable to be updated.
	Value	The new value.

Example

```
uSetUserVar("MyVar", "NewText");
```

9.1.18.123 uSHA1

ⓘ Note

As of SAP NetWeaver Identity Management Identity Center 7.2, it is recommended to use the function uGenHash instead.

Description: Returns the SHA1 checksum for a string.

Syntax: OutString=uSHA1(String Input[, String CharacterEncoding]);

Parameters:	Input	Original string.
	CharacterEncoding	Optional. Specifies which character set to use, and must be the name of a supported character set. Which character sets are supported varies from platform to platform, but every implementation of the Java platform is required to support the following standard character sets: • US-ASCII • ISO-8859-1 • UTF-8 • UTF-16BE • UTF-16LE • UTF-16 Other character sets may be available, but this requires a manual installation. If this parameter is not specified, the default character set for the platform is used.
	OutString	Returned checksum.

Example

```
InString = "Madame"  
OutString = uSHA1(InString);  
' --- OutString = <Checksum>
```

9.1.18.124 uShell

Description: Executes a shell command.

Syntax: uShell(String Command, long ViewFlags, long Flags);

Parameters: **Command** This is the command to be executed. It can for example, be used to start an application. This command should be written in the same way as if typed in a DOS window.

ViewFlags 0: Hide application
1: Normal window
2: Minimized window
3: Maximized window

Flags 0: No wait
2: Wait for execution

9.1.18.125 uShellRead

Description: Executes a shell command and reads the output.

Syntax: OutString=uShellRead(String Command, Int Flags);

Parameters: **Command** The command to be executed.

Flags Specifies what output to read.
Possible values are:
0: Read all
1: Read stdout
2: Read stderr

OutString The output from the command.

Example

```
// Main function: CheckDiskSpace
/* On a unix system, the df command will give an output like this:
Filesystem            1K-blocks      Used Available Use% Mounted on
/dev/hda2              3510364    1777740    1554304   54% /
/dev/hda1              101089      9484       86386   10% /boot
none                  127692         0      127692    0% /dev/shm
```

The following unix script (stored in /home/dse/dfcheck.sh) will return just the percent used:

```
df | grep $1 | awk '{print $5}' | awk -F % '{print $1}'
*/

function CheckDiskSpace(Par)
{
    /* Execute shell command and read output
    * (Flags=0 is read all, 1 is only read stdout, 2 is only read stderr)
    */
    aRes = UserFunc.uShellRead("/home/dse/dfcheck.sh " + Par, 0);
    // We need at least 10 percent free
    if (aRes > 90 )
    {
        UserFunc.uErrMsg(2,"High Disk Usage : " + aRes);
    }
    else
    {
        UserFunc.uErrMsg(0,"Normal Disk Usage : " + aRes);
    }
}
```

9.1.18.126 uSkip

Description: Skips current entry or current pass.

Syntax: uSkip(Int OperationType[, Int ProvCommit] [, String Msg]);

Parameters: **OperationType** Specifies what to skip.

1: Skip entry

2: Skip pass

ProvCommit Optional. Only relevant in provisioning jobs. Specifies the status to be set when the job completes:

1: Set provisioning status OK (Default)

2: Set provisioning status to failed

Msg Optional. Message text to add to the provisioning status.

Example

```
OperationType = 1  
uSkip(OperationType);
```

Skips the current entry.

9.1.18.127 uSleep

Description: Delays the execution of the job for the specified number of milliseconds.

Syntax: uSleep(long Milliseconds);

Parameters: Milliseco Number of milliseconds.
nds

Example

Sample Code

```
' Holds execution for the number of  
' milliseconds given by 'par'  
Function Tired(Par)  
  call uSleep(cLng(par))  
end function  
  
Tired(1000)
```

The execution of the job is delayed for 1 second.

9.1.18.128 uSplitString

Description: Split a string on a given delimiter. Returns a Vector of String.

Note

In JavaScript there is a split method on strings, but this may give different results depending on which type of string object you have. We recommend using the uSplitString function.

Syntax: Vector=uSplitString(String Input, String Delimiter [,String QuoteCharacter]);

Parameters: Input String to split.

Delimiter	String to use as delimiter. Default delimiter is "," (comma).
QuoteCharacter	Optional. Of QuoteCharacter is specified, Delimiter must be a single character. Two quotes are treated as a single character.
Vector	A vector of String.

Example

```
Vector = uSplitString("Hello World", " ");
/* vector will now contain to elements "Hello" and "World!" */
```

9.1.18.129 uStop

Description:	Stops executing the job after it has processed the current entry. To stop the job immediately without completing the current entry, call <i>uSkip</i> first.
Syntax:	uStop(String LogMessage);
Parameters:	LogMessage Message to write to the error log.

Example

```
LogMessage = "Job stopped because of too many errors"
uStop(LogMessage);
```

This function stops the job and adds a message to the error log.

9.1.18.130 uStrConv

Description:	Converts a string to upper case, lower case or mixed case. Useful for converting names of persons to a consistent format.
Syntax:	OutString=uStrConv(String InString, long Operation [,long CodePage]);
Parameters:	InString String to convert.

Operation	Conversion type: 1: Convert to lower case 2: Convert to upper case 3: First letter in each word in upper case, the rest in lower case.
CodePage	Optional. Code page to use.
OutString	Returned string.

Example

```
Function Nice(Par){
' If Par= oLe olSEn
' the result will be
' Ole Olsen
return uStrConv(cstr(par),3);
}
```

9.1.18.131 uToBase64

Description:	Converts a string to Base64.
Syntax:	Base64String=uToBase64(String InString[, String CharacterEncoding]);
Parameters:	InString Original string.

CharacterEncoding

Note

Only available with the Java runtime engine.

Optional. Specifies which character set to use when converting the string, and must be the name of a supported character set.

Which character sets are supported varies from platform to platform, but every implementation of the Java platform is required to support the following standard character sets:

- US-ASCII
- ISO-8859-1
- UTF-8
- UTF-16BE
- UTF-16LE
- UTF-16

Other character sets may be available, but this requires a manual installation.

If this parameter is not specified, the default character set for the platform is used.

Base64String	Returned Base64 encoded string.

Example

```
InString = "Madame"  
Base64String=uToBase64(InString)  
' --- Base64String = <Base64 encoded string>
```

9.1.18.132 uToBinFile

Description: Writes binary data, such as pictures or certificates, to a file.

Syntax: uToBinFile(String Filename, String Data[, String CharacterEncoding]);

Parameters: **Filename** Name of file to write to.

CharacterEncoding

Note

Only available with the Java runtime engine.

Optional. Specifies which character set to use, and must be the name of a supported character set.

Which character sets are supported varies from platform to platform, but every implementation of the Java platform is required to support the following standard character sets:

- US-ASCII
- ISO-8859-1
- UTF-8
- UTF-16BE
- UTF-16LE
- UTF-16

Other character sets may be available, but this requires a manual installation.

If this parameter is not specified, the default character set for the platform is used.

Data	Data to store. This can be binary data, like a certificate or JPEG picture.

Example

```
Function StoreData(Par){
  uToBinFile("d:\temp\tofile.bin",cstr(par));
}
```

9.1.18.133 uToFile

Description:	Writes data to a file.	
Syntax:	uToFile(String FileName, String WriteText, Boolean AppendToFile[, String CharacterEncoding]);	
Parameters:	FileName	Name of file to write to.
	WriteText	Text to write.
	AppendToFile	True: Append to file
	le	False: Write from start of file

CharacterEncoding

Note

Only available with the Java runtime engine.

Optional. Specifies which character set to use when converting the decoded bytes to the returned string, and must be the name of a supported character set.

Which character sets are supported varies from platform to platform, but every implementation of the Java platform is required to support the following standard character sets:

- US-ASCII
- ISO-8859-1
- UTF-8
- UTF-16BE
- UTF-16LE
- UTF-16

Other character sets may be available, but this requires a manual installation.

If this parameter is not specified, the default character set for the platform is used.

Example

```
FileName = "%$ddm.ddmpath%\tofile.txt"
WriteText = "Hello" & vbCrLf & "world"
AppendToFile = FALSE
call uToFile(FileName,WriteText,bAppend)
```

This function will store the text "Hello" in line one and "world" in line two in a file stored in the directory where the current job is located.

9.1.18.134 uToHex

Description: Converts a string to hex code. This function is often used for representing binary data as standard ASCII text.

Syntax: HexString=uToHex(String InString[, String CharacterEncoding]);

Parameters: InString Original string.

CharacterEncoding

Note

Only available with the Java runtime engine.

Optional. Specifies which character set to use when converting the string, and must be the name of a supported character set.

Which character sets are supported varies from platform to platform, but every implementation of the Java platform is required to support the following standard character sets:

- US-ASCII
- ISO-8859-1
- UTF-8
- UTF-16BE
- UTF-16LE
- UTF-16

Other character sets may be available, but this requires a manual installation.

If this parameter is not specified, the default character set for the platform is used.

HexString	Returned hex value.

Example

```
InString = "Madame"  
HexString=uToHex(InString)  
' --- HexString = <hex encoded string>
```

The result will be the double length of input string.

```
InString = "Madame"  
HexString=uToHex(InString, "ISO-8859-1")  
' --- HexString = <hex encoded string>
```

9.1.18.135 uWarning

Description: Writes a warning to the log. Substitutes *uErrMsg*.

Note

If tracing is enabled, any messages written by this function will also be included in the trace.

Syntax:	uWarning(String Msg);

Parameters:	Msg A message string.

Example

```
uWarning("My warning");
```

9.2 Appendix

9.2.1 Repository Constants

You can find the repository constants for repository definitions used by SAP provisioning framework in the *Related Information* section.

Related Information

[Repository Constants for AS ABAP \(Load Balanced Connection\) \[page 1235\]](#)

[Repository Constants for AS ABAP \(Specific Application Server\) \[page 1238\]](#)

[Repository Constants for AS Java \(LDAP Backend\) \[page 1241\]](#)

[Repository Constants for AS Java \[page 1243\]](#)

[Repository Constants for a Dual-Stack System \(Load Balanced Connection\) \[page 1244\]](#)

[Repository Constants for a Dual-Stack System \(Specific Application Server\) \[page 1248\]](#)

[Repository Constants for Business Suite AS ABAP \[page 1251\]](#)

[Repository Constants for SUN \[page 1255\]](#)

[Repository Constants for Active Directory for Provisioning Framework \[page 1256\]](#)

[Repository Constants for SAP HANA Database \[page 1259\]](#)

[Repository Constants for SuccessFactors \[page 1261\]](#)

[Repository Constants for Lotus Notes \[page 1263\]](#)

[Repository Constants for Identity Authentication System \[page 1267\]](#)

[Repository Constants for a SCIM System \[page 1269\]](#)

[Repository Constants for SAP S/4HANA \[page 1274\]](#)

[Repository Types, Constants and Variables \[page 101\]](#)

9.2.1.1 Repository Constants for AS ABAP (Load Balanced Connection)

Repository Constants

Repository Constant	Value
JCO_CLIENT_MSHOST	<message_server_host> Host name or IP address of the message server
JCO_CLIENT_R3NAME	<SID> AS ABAP System ID
JCO_CLIENT_GROUP	<logon group>, for example, PUBLIC The name of AS ABAP logon group
JCO_CLIENT_USER	<user_ID> User name of the communication (technical) user
JCO_CLIENT_PASSWD	<password> Password of the communication (technical) user For AS ABAP Release 4.6C, use uppercase.
JCO_CLIENT_CLIENT	<client> Client of the communication (technical) user
JCO_CLIENT_LANG	<language ID>, for example, EN Language of the communication (technical) user
JCO_CLIENT_ASHOST	Not applicable Host name or IP address of a specific AS ABAP
JCO_CLIENT_GWHOST	<gateway host> AS ABAP Gateway Host To find out the data for your Gateway, choose Administration > System Administration > Monitor > System Monitoring > Gateway Monitor. A list is displayed of all the SAP Gateways connected to your SAP System. Usually, an Gateway is assigned to each SAP application server and database server.

Repository Constant	Value
JCO_CLIENT_GWSERV	<gateway service> (for example, "sapgw00") AS ABAP Gateway Service
JCO_CLIENT_SYSNR	<system number> AS ABAP system number
JCO_CLIENT_SNC_LIB	<Path to library which provides SNC service>
JCO_CLIENT_SNC_MODE	<Secure network connection (SNC) mode> Possible values: • 0 (off) • 1 (on)
JCO_CLIENT_SNC_MYNAME	<SNC name> SNC name; overrides default SNC partner
JCO_CLIENT_SNC_PARTNERNAME	<SNC partner> SNC partner; e.g. p:CN=R3, O=XYZ-INC, C=EN
JCO_CLIENT_SNC_QOP	<SNC level of security> Possible values: from 1 to 9
CUA_MASTER	<TRUE/FALSE> Defines if system is the Central System of Central User Administration Possible values: • TRUE - Composite roles are read into the identity store without their child privileges. • FALSE - Composite roles are read into the identity store with their child privileges.
MX_PRIV_GROUPING_ATTRIBUTE	MX_IS_ACCOUNT
MX_PRIV_GROUPING_RULE	P:4 With this value, entries are grouped if they have the same attribute defined in the repository constant MX_PRIV_GROUPING_ATTRIBUTE.

Repository Constant	Value
SYSTEM_PRIVILEGE	PRIV:SYSTEM:<Repository> <Repository> is the exact name of the repository to which the constants belong. In SAP Identity Management Administration User Interface, the value of the SYSTEM_PRIVILEGE repository constant is read only. When a repository is created, the name of the repository is automatically filled in.

The following constants are repository type constants. They are valid for all repositories of this type and visible only in SAP Identity Management Developer Studio.

Repository Type Constants

Repository Type Constant	Value
MX_ADD_MEMBER_TASK	<process number for Provisioning>
MX_DEL_MEMBER_TASK	<process number for Deprovisioning>
MX_MODIFYTASK	<process number for Modify>
MX_OFFSET_ADD_MEMBER	<0>: On ValidFrom When to execute the Add member process
MX_REQ_PRIV_PCYADD_MISSING	<1>: Wait, that is wait for the privilege to be assigned Policy when the master privilege is missing
MX_REQ_PRIV_TIMEOUT	<0> Number of seconds to wait before timeout of the Wait policy
MX_REQ_PRIV	<Privilege reference (MSKEYVALUE)> Master privilege
REPOSITORY_SYNC	SYNC
REPOSITORY_TYPE	ABAP Repository Type
INITIAL_LOAD	<Initial Load Job>
READ_HELP_VALUES	<Read Help Values Job>
RESET_DELTA	<Reset Delta Job>

For more information about configuring Secure Network Communications (SNC), see Appendix D: Configuring the ABAP Connector to use SNC.

Related Information

[Configuring the ABAP Connector to Use SNC \[page 1313\]](#)

9.2.1.2 Repository Constants for AS ABAP (Specific Application Server)

Repository Constants

Repository Constant	Value
JCO_CLIENT_MSHOST	Not applicable Host name or IP address of the message server
JCO_CLIENT_R3NAME	<SID> AS ABAP System ID
JCO_CLIENT_GROUP	Not applicable The name of AS ABAP logon group
JCO_CLIENT_USER	<user_ID> User name of the communication (technical) user
JCO_CLIENT_PASSWD	<password> Password of the communication (technical) user For AS ABAP Release 4.6C, use uppercase.
JCO_CLIENT_CLIENT	<client> Client of the communication (technical) user
JCO_CLIENT_LANG	<language ID>, for example, EN Language of the communication (technical) user
JCO_CLIENT_ASHOST	<application_server_host> Host name or IP address of a specific AS ABAP

Repository Constant	Value
JCO_CLIENT_GWHOST	<gateway host> AS ABAP Gateway Host To find out the data for your Gateway, choose Administration > System Administration > Monitor > System Monitoring > Gateway Monitor. A list is displayed of all the SAP Gateways connected to your SAP System. Usually, an Gateway is assigned to each SAP application server and database server.
JCO_CLIENT_GWSERV	<gateway service> (for example, "sapgw00") AS ABAP Gateway Service
JCO_CLIENT_SYSNR	<system number> AS ABAP system number
JCO_CLIENT_SNC_LIB	<Path to library which provides SNC service>
JCO_CLIENT_SNC_MODE	<Secure network connection (SNC) mode> Possible values: • 0 (off) • 1 (on)
JCO_CLIENT_SNC_MYNAME	<SNC name> SNC name; overrides default SNC partner
JCO_CLIENT_SNC_PARTNERNAME	<SNC partner> SNC partner, e.g. p:CN=R3, O=XYZ-INC, C=EN
JCO_CLIENT_SNC_QOP	<SNC level of security> Possible values: from 1 to 9
CUA_MASTER	<TRUE/FALSE> Defines if system is the Central System of Central User Administration Possible values: • TRUE - Composite roles are read into the identity store without their child privileges. • FALSE - Composite roles are read into the identity store with their child privileges.

Repository Constant	Value
MX_PRIV_GROUPING_ATTRIBUTE	MX_IS_ACCOUNT
MX_PRIV_GROUPING_RULE	P:4 With this value, entries are grouped if they have the same attribute defined in the repository constant MX_PRIV_GROUPING_ATTRIBUTE.
SYSTEM_PRIVILEGE	PRIV:SYSTEM:<Repository> <Repository> is the exact name of the repository to which the constants belong. In SAP Identity Management Administration User Interface, the value of the SYSTEM_PRIVILEGE repository constant is read only. When a repository is created, the name of the repository is automatically filled in.

The following constants are repository type constants. They are valid for all repositories of this type and visible only in SAP Identity Management Developer Studio.

Repository Type Constants

Repository Type Constant	Value
MX_ADD_MEMBER_TASK	<process number for Provisioning>
MX_DEL_MEMBER_TASK	<process number for Deprovisioning>
MX_MODIFYTASK	<process number for Modify>
MX_OFFSET_ADD_MEMBER	<0>: On ValidFrom When to execute the Add member process
MX_REQ_PRIV_PCYADD_MISSING	<1>: Wait, that is wait for the privilege to be assigned Policy when the master privilege is missing
MX_REQ_PRIV_TIMEOUT	<0> Number of seconds to wait before timeout of the Wait policy
MX_REQ_PRIV	<Privilege reference (MSKEYVALUE)> Master privilege
REPOSITORY_SYNC	SYNC
REPOSITORY_TYPE	ABAP Repository Type
INITIAL_LOAD	<Initial Load Job>

Repository Type Constant	Value
READ_HELP_VALUES	<Read Help Values Job>
RESET_DELTA	<Reset Delta Job>

For more information about configuring Secure Network Communications (SNC), see Appendix D: Configuring the ABAP Connector to use SNC.

Related Information

[Configuring the ABAP Connector to Use SNC \[page 1313\]](#)

9.2.1.3 Repository Constants for AS Java (LDAP Backend)

Repository Constants

Repository Constant	Value
HTTP_Protocol	<http/https> HTTP Protocol used for connection (http or https). For password provisioning SSL must be set up: <TRUE/FALSE>
APPLICATION_HOST	<host> AS Java host name or IP address
HTTP_PORT	<http_port> AS Java HTTP port
HTTP_AUTH_USER	<user_ID> User name of the communication (technical) user
HTTP_AUTH_PWD	<password> Password of the communication (technical) user
BACKEND_REPOSITORYNAME	<LDAP directory repository name> Repository name of the LDAP backend repository
BACKEND_TYPE	LDAP Backend Type

Repository Constant	Value
SYSTEM_PRIVILEGE	PRIV:SYSTEM:<Repository> <Repository> is the exact name of the repository to which the constants belong. In SAP Identity Management Administration User Interface, the value of the SYSTEM_PRIVILEGE repository constant is read only. When a repository is created, the name of the repository is automatically filled in.
MX_PRIV_GROUPING_ATTRIBUTE	MX_PRIVILEGE_TYPE
MX_PRIV_GROUPING_RULE	P:5 With this value, entries are grouped if they have the same attribute defined in the repository constant MX_PRIV_GROUPING_ATTRIBUTE and if they are of the same operation type, for example, add or remove.

The following constants are repository type constants. They are valid for all repositories of this type and visible only in SAP Identity Management Developer Studio.

Repository Type Constants

Repository Type Constant	Value
MX_ADD_MEMBER_TASK	<process number for Provisioning>
MX_DEL_MEMBER_TASK	<process number for Deprovisioning>
MX_MODIFYTASK	<process number for Modify>
MX_OFFSET_ADD_MEMBER	<0>: On ValidFrom When to execute the Add member process
MX_REQ_PRIV_PCYADD_MISSING	<1>: Wait, that is wait for the privilege to be assigned Policy when the master privilege is missing
MX_REQ_PRIV_TIMEOUT	<0> Number of seconds to wait before timeout of the Wait policy
MX_REQ_PRIV	<Privilege reference (MSKEYVALUE)> Master privilege
REPOSITORY_SYNC	SYNC
REPOSITORY_TYPE	JAVA Repository Type

Repository Type Constant	Value
INITIAL_LOAD	<Initial Load Job>

9.2.1.4 Repository Constants for AS Java

Repository Constants

Repository Constant	Value
HTTP_Protocol	<http/https> HTTP Protocol used for connection (http or https) For password provisioning SSL must be set up.<TRUE/FALSE>
APPLICATION_HOST	<host> AS Java host name or IP address
HTTP_PORT	<http_port> AS Java HTTP port
HTTP_AUTH_USER	<user_ID> User name of the communication (technical) user
HTTP_AUTH_PWD	<password> Password of the communication (technical) user
SYSTEM_PRIVILEGE	PRIV:SYSTEM:<Repository> <Repository> is the exact name of the repository to which the constants belong. In SAP Identity Management Administration User Interface, the value of the SYSTEM_PRIVILEGE repository constant is read only. When a repository is created, the name of the repository is automatically filled in.
MX_PRIV_GROUPING_ATTRIBUTE	MX_PRIVILEGE_TYPE
MX_PRIV_GROUPING_RULE	P:5 With this value, entries are grouped if they have the same attribute defined in the repository constant MX_PRIV_GROUPING_ATTRIBUTE and if they are of the same operation type, for example, add or remove.

The following constants are repository type constants. They are valid for all repositories of this type and visible only in SAP Identity Management Developer Studio.

Repository Type Constants

Repository Type Constant	Value
MX_ADD_MEMBER_TASK	<process number for Provisioning>
MX_DEL_MEMBER_TASK	<process number for Deprovisioning>
MX_MODIFYTASK	<process number for Modify>
MX_OFFSET_ADD_MEMBER	<0>: On ValidFrom When to execute the Add member process
MX_REQ_PRIV_PCYADD_MISSING	<1>: Wait, that is wait for the privilege to be assigned Policy when the master privilege is missing
MX_REQ_PRIV_TIMEOUT	<0> Number of seconds to wait before timeout of the Wait policy
MX_REQ_PRIV	<Privilege reference (MSKEYVALUE)> Master privilege
BACKEND_TYPE	DB Backend type
REPOSITORY_SYNC	SYNC
INITIAL_LOAD	<Initial Load Job>

9.2.1.5 Repository Constants for a Dual-Stack System (Load Balanced Connection)

Repository Constants

Repository Constant	Value
JCO_CLIENT_ASHOST	Not applicable Host name or IP address of a specific AS ABAP
JCO_CLIENT_USER	<user_ID> AS ABAP user name

Repository Constant	Value
JCO_CLIENT_PASSWD	< password> AS ABAP password For AS ABAP Release 4.6C, use uppercase.
JCO_CLIENT_MSHOST	<message_server_host> Host name or IP address of the message server
JCO_CLIENT_R3NAME	<SID> AS ABAP System ID
JCO_CLIENT_GROUP	<AS ABAP logon group> for example, PUBLIC The name of AS ABAP logon group
JCO_CLIENT_CLIENT	<client> Client of the communication (technical) user
JCO_CLIENT_GWHOST	<gateway host> AS ABAP Gateway Host To find out the data for your Gateway, choose Administration > System Administration > Monitor > System Monitoring > Gateway Monitor . A list is displayed of all the SAP Gateways connected to your SAP System. Usually, an Gateway is assigned to each SAP application server and database server.
JCO_CLIENT_GWSERV	<gateway service> (for example, "sapgw00") AS ABAP Gateway Service
JCO_CLIENT_LANG	<language ID>, for example, EN Language of the communication (technical) user
JCO_CLIENT_SYSNR	<system number>
HTTP_PROTOCOL	<http/https> HTTP Protocol used for connection (http or https)
APPLICATION_HOST	<host> AS Java host name or IP address

Repository Constant	Value
HTTP_PORT	AS Java http_port AS Java HTTP port
HTTP_AUTH_USER	<user_ID> AS Java user name
HTTP_AUTH_PWD	<password> AS Java password
JCO_CLIENT_SNC_LIB	<Path to library which provides SNC service>
JCO_CLIENT_SNC_MODE	<Secure network connection (SNC) mode> Possible values: • 0 (off) • 1 (on)
JCO_CLIENT_SNC_MYNAME	<SNC name> SNC name; overrides default SNC partner.
JCO_CLIENT_SNC_PARTNERNAME	<SNC partner> SNC partner; e.g. p:CN=R3, O=XYZ-INC, C=EN
JCO_CLIENT_SNC_QOP	<SNC level of security> Possible values: from 1 to 9
SYSTEM_PRIVILEGE	PRIV:SYSTEM:<Repository> <Repository> is the exact name of the repository to which the constants belong. In SAP Identity Management Administration User Interface, the value of the SYSTEM_PRIVILEGE repository constant is read only. When a repository is created, the name of the repository is automatically filled in.
CUA_MASTER	<TRUE/FALSE> Defines if system is the Central System of Central User Administration Possible values: • TRUE - Composite roles are read into the identity store without their child privileges. • FALSE - Composite roles are read into the identity store with their child privileges.

Repository Constant	Value
MX_PRIV_GROUPING_ATTRIBUTE	MX_IS_ACCOUNT
MX_PRIV_GROUPING_RULE	P:4 With this value, entries are grouped if they have the same attribute defined in the repository constant MX_PRIV_GROUPING_ATTRIBUTE.

The following constants are repository type constants. They are valid for all repositories of this type and visible only in SAP Identity Management Developer Studio.

Repository Type Constants

Repository Type Constant	Value
MX_ADD_MEMBER_TASK	<process number for Provisioning>
MX_DEL_MEMBER_TASK	<process number for Deprovisioning>
MX_MODIFYTASK	<process number for Modify>
MX_OFFSET_ADD_MEMBER	<0>: On ValidFrom When to execute the Add member process
MX_REQ_PRIV_PCYADD_MISSING	<1>: Wait, that is wait for the privilege to be assigned Policy when the master privilege is missing
MX_REQ_PRIV_PCYADD_MISSING	<0> Number of seconds to wait before timeout of the Wait policy
MX_REQ_PRIV	<Privilege reference (MSKEYVALUE)> Master privilege
REPOSITORY_SYNC	SYNC
REPOSITORY_TYPE	DUALABAP Repository type

For more information about configuring Secure Network Communications (SNC), see Appendix D: Configuring the ABAP Connector to use SNC.

Related Information

[Configuring the ABAP Connector to Use SNC \[page 1313\]](#)

9.2.1.6 Repository Constants for a Dual-Stack System (Specific Application Server)

Repository Constants

Repository Constant	Value
JCO_CLIENT_USER	<user_ID> User name of the communication (technical) user
JCO_CLIENT_ASHOST	<application_server_host> Host name or IP address of a specific AS ABAP
JCO_CLIENT_SYSNR	<system number> AS ABAP system number
JCO_CLIENT_PASSWD	<password> Password of the communication (technical) user For AS ABAP Release 4.6C, use uppercase.
JCO_CLIENT_CLIENT	<client> Client of the communication (technical) user
JCO_CLIENT_LANG	<language ID>, for example, EN Language of the communication (technical) user
JCO_CLIENT_GROUP	Not applicable The name of AS ABAP logon group
JCO_CLIENT_GWHOST	<gateway host> AS ABAP Gateway Host To find out the data for your Gateway, choose Administration > System Administration > Monitor > System Monitoring > Gateway Monitor. A list is displayed of all the SAP Gateways connected to your SAP System. Usually, an Gateway is assigned to each SAP application server and database server.
JCO_CLIENT_GWSERV	<gateway service> (for example, "sapgw00") AS ABAP Gateway Service

Repository Constant	Value
JCO_CLIENT_MSHOST	Not applicable Host name or IP address of the message server
JCO_CLIENT_R3NAME	<SID> AS ABAP System ID
JCO_CLIENT_SNC_LIB	<Path to library which provides SNC service>
JCO_CLIENT_SNC_MODE	<Secure network connection (SNC) mode> Possible values: • 0 (off) • 1 (on)
JCO_CLIENT_SNC_MYNAME	<SNC name> SNC name; overrides default SNC partner.
JCO_CLIENT_SNC_PARTNERNAME	<SNC partner> SNC partner; e.g. p:CN=R3, O=XYZ-INC, C=EN
JCO_CLIENT_SNC_QOP	<SNC level of security> Possible values: from 1 to 9
HTTP_PROTOCOL	<http/https> HTTP Protocol used for connection (http or https)
APPLICATION_HOST	<host> AS Java host name or IP address
HTTP_PORT	<http_port> AS Java HTTP port
HTTP_AUTH_USER	<user_ID> AS Java user name
HTTP_AUTH_PWD	<password> AS Java password

Repository Constant	Value
CUA_MASTER	<TRUE/FALSE> Defines if system is the Central System of Central User Administration Possible values: - TRUE - Composite roles are read into the identity store without their child privileges. - FALSE - Composite roles are read into the identity store with their child privileges.
SYSTEM_PRIVILEGE	PRIV:SYSTEM:<Repository> <Repository> is the exact name of the repository to which the constants belong. In SAP Identity Management Administration User Interface, the value of the SYSTEM_PRIVILEGE repository constant is read only. When a repository is created, the name of the repository is automatically filled in.
MX_PRIV_GROUPING_ATTRIBUTE	MX_IS_ACCOUNT
MX_PRIV_GROUPING_RULE	P:4 With this value, entries are grouped if they have the same attribute defined in the repository constant MX_PRIV_GROUPING_ATTRIBUTE.

The following constants are repository type constants. They are valid for all repositories of this type and visible only in SAP Identity Management Developer Studio.

Repository Type Constants

Repository Type Constant	Value
MX_ADD_MEMBER_TASK	<process number for Provisioning>
MX_DEL_MEMBER_TASK	<process number for Deprovisioning>
MX_MODIFYTASK	<process number for Modify>
MX_OFFSET_ADD_MEMBER	<0>: On ValidFrom When to execute the Add member process
MX_REQ_PRIV_PCYADD_MISSING	<1>: Wait, that is wait for the privilege to be assigned Policy when the master privilege is missing

Repository Type Constant	Value
MX_REQ_PRIV_TIMEOUT	<0> Number of seconds to wait before timeout of the Wait policy
MX_REQ_PRIV	<Privilege reference (MSKEYVALUE)> Master privilege
REPOSITORY_SYNC	SYNC
REPOSITORY_TYPE	DUALABAP Repository type

For more information about configuring Secure Network Communications (SNC), see Appendix D: Configuring the ABAP Connector to use SNC.

Related Information

[Configuring the ABAP Connector to Use SNC \[page 1313\]](#)

9.2.1.7 Repository Constants for Business Suite AS ABAP

Repository Constants

Repository Constant	Value
JCO_CLIENT_MSHOST	- For load balancing: <message_server_host> Host name or IP address of the message server - For specific application server connection: Not applicable
JCO_CLIENT_R3NAME	<SID> AS ABAP System ID
JCO_CLIENT_GROUP	- For load balancing: <logon group> The name of AS ABAP logon group - For specific application server connection: Not applicable

Repository Constant	Value
JCO_CLIENT_USER	<user_ID> User name of the communication (technical) user
JCO_CLIENT_PASSWD	<password> Password of the communication (technical) user For AS ABAP Release 4.6C, use uppercase.
JCO_CLIENT_CLIENT	<client> Client of the communication (technical) user
JCO_CLIENT_LANG	<language ID>, for example, EN Language of the communication (technical) user
CUA_MASTER	<TRUE/FALSE> Defines if system is the Central System of Central User Administration Possible values: - TRUE - Composite roles are read into the identity store without their child privileges. - FALSE - Composite roles are read into the identity store with their child privileges.
JCO_CLIENT_ASHOST	- For load balancing: Not applicable - For specific application server connection: <application_server_host> Host name or IP address of a specific AS ABAP
JCO_CLIENT_GWHOST	<gateway host> AS ABAP Gateway Host To find out the data for your Gateway, choose Administration > System Administration > Monitor > System Monitoring > Gateway Monitor. A list is displayed of all the SAP Gateways connected to your SAP System. Usually, an Gateway is assigned to each SAP application server and database server.
JCO_CLIENT_GWSERV	<gateway service> (for example, "sapgw00") AS ABAP Gateway Service

Repository Constant	Value
JCO_CLIENT_SYSNR	<system number> AS ABAP system number
NO_USER_ACCOUNT	0 Defines for a business suite system if an ABAP user is created or only the business suite functionality is called. Default value: 0 Possible values: 0 (This means that the connector creates ABAP users and business suite objects for example, employees.) 1 (This means that you do not want a user account created in the target system for an identity.)
JCO_CLIENT_SNC_LIB	<Path to library which provides SNC service>
JCO_CLIENT_SNC_MODE	<Secure network connection (SNC) mode> Possible values: 0 (off) 1 (on)
JCO_CLIENT_SNC_MYNAME	<SNC name> SNC name; overrides default SNC partner.
JCO_CLIENT_SNC_PARTNERNAME	<SNC partner> SNC partner; e.g. p:CN=R3, O=XYZ-INC, C=EN
JCO_CLIENT_SNC_QOP	<SNC level of security> Possible values: from 1 to 9
MX_PRIV_GROUPING_ATTRIBUTE	MX_IS_ACCOUNT
MX_PRIV_GROUPING_RULE	P:4 With this value, entries are grouped if they have the same attribute defined in the repository constant MX_PRIV_GROUPING_ATTRIBUTE.

Repository Constant	Value
SYSTEM_PRIVILEGE	PRIV:SYSTEM:<Repository> <Repository> is the exact name of the repository to which the constants belong. In SAP Identity Management Administration User Interface, the value of the SYSTEM_PRIVILEGE repository constant is read only. When a repository is created, the name of the repository is automatically filled in.

The following constants are repository type constants. They are valid for all repositories of this type and visible only in SAP Identity Management Developer Studio.

Repository Type Constants

Repository Type Constant	Value
MX_ADD_MEMBER_TASK	<process number for Provisioning>
MX_DEL_MEMBER_TASK	<process number for Deprovisioning>
MX_MODIFYTASK	<process number for Modify>
MX_OFFSET_ADD_MEMBER	<0>: On ValidFrom When to execute the Add member process
MX_REQ_PRIV_PCYADD_MISSING	<1>: Wait, that is wait for the privilege to be assigned Policy when the master privilege is missing
MX_REQ_PRIV_TIMEOUT	<0> Number of seconds to wait before timeout of the Wait policy
MX_REQ_PRIV	<Privilege reference (MSKEYVALUE)> Master privilege
REPOSITORY_SYNC	SYNC
REPOSITORY_TYPE	ABAP Repository type
INITIAL_LOAD	<Initial Load Job>
RESET_DELTA	<Reset Delta Job>
PROVISION_COMPANY_ADDRESS_TO_ABAP	<Provision Company Address to ABAP Job>

For more information about configuring Secure Network Communications (SNC), see Appendix D: Configuring the ABAP Connector to use SNC.

Related Information

[Configuring the ABAP Connector to Use SNC \[page 1313\]](#)

9.2.1.8 Repository Constants for SUN

Repository Constants

Repository Constant	Value
LDAP_HOST	<host> Directory server host name or IP address
LDAP_STARTING_POINT	<LDAP starting point> Directory server starting point
LDAP_STARTING_POINT_GROUPS	<LDAP starting point for groups> Directory server starting point for groups
LDAP_PORT	<LDAP port>
LDAP_LOGIN	<LDAP user ID>
LDAP_PASSWORD	<password>
SYSTEM_PRIVILEGE	PRIV:SYSTEM:<Repository> <Repository> is the exact name of the repository to which the constants belong. In SAP Identity Management Administration User Interface, the value of the SYSTEM_PRIVILEGE repository constant is read only. When a repository is created, the name of the repository is automatically filled in.

The following constants are repository type constants. They are valid for all repositories of this type and visible only in SAP Identity Management Developer Studio.

Repository Type Constants

Repository Type Constant	Value
MX_ADD_MEMBER_TASK	<process number for Provisioning>
MX_DEL_MEMBER_TASK	<process number for Deprovisioning>
MX_MODIFYTASK	<process number for Modify>

Repository Type Constant	Value
MX_OFFSET_ADD_MEMBER	<0>: On ValidFrom When to execute the Add member process
MX_REQ_PRIV	<Privilege reference (MSKEYVALUE)> Master privilege
REPOSITORY_SYNC	SYNC
REPOSITORY_TYPE	SUN Repository type
INITIAL_LOAD	<Initial Load Job>

9.2.1.9 Repository Constants for Active Directory for Provisioning Framework

Repository Constants

Repository Constant	Value
LDAP_HOST	<Host name of ADS>
LDAP_STARTING_POINT	<LDAP starting point>
LDAP_STARTING_POINT_GROUPS	<LDAP starting point for groups>
LDAP_PORT	<LDAP port>
LDAP_PORT_SSL	<LDAP port> An SSL connection is used for password provisioning. ¹ For password provisioning SSL must be set up.
LDAP_LOGIN	<LDAP user ID>
LDAP_PASSWORD	<password>
LDAP_FILTER	(objectclass=person) LDAP filter for entries
LDAP_FILTER_GROUPS	(objectclass=group) LDAP filter for groups

Repository Constant	Value
SYSTEM_PRIVILEGE	PRIV:SYSTEM:<Repository> <Repository> is the exact name of the repository to which the constants belong. In SAP Identity Management Administration User Interface, the value of the SYSTEM_PRIVILEGE repository constant is read only. When a repository is created, the name of the repository is automatically filled in.

1

To set up an SSL/TLS secured connection between the Active Directory Server and the Identity Center, you set this constant to the repository value of the Active Directory Server and install the necessary certificates. For more information, see section *Identity Center: Repository Security* of the *SAP Identity Management Security Guide*.

The following constants are only necessary if you want to create or change Exchange mailboxes.

Optional Repository Constants

Repository Constant	Value	Comment
EXCHANGE_VERSION	Default value: <Empty string> Possible values: • <Empty string> (An empty value indicates that no Exchange user should be created.) • <Exchange2010> (Set this value if Exchange 2010 is used, and an Exchange user should be created.) • <Exchange2007> (Set this value if Exchange 2007 is used, and an Exchange user should be created.)	By setting any value for this attribute, you define whether existing ADS users are assigned further attributes for using Exchange. That is, you activate the use of Exchange provisioning. You also indicate your Exchange version.
LDAP_MAIL_DOMAIN	<Mail domain>	LDAP Mail Domain name
LDAP_UPN	<LDAP UPN>	LDAP User principal name
LDAP_TEMPLATE_MAILBOX	<DN of the template mailbox>	The distinguished name of the template mailbox, e.g. CN=Template1,CN=Users,DC=ex2k7,DC=dom
LDAP_TEMPLATE_MAILBOX_ATTRIBUTES_2007	<Template mailbox attributes for Exchange 2007>	
LDAP_TEMPLATE_MAILBOX_ATTRIBUTES_2010	<Template mailbox attributes for Exchange 2010>	

The following constants are repository type constants. They are valid for all repositories of this type and visible only in SAP Identity Management Developer Studio.

Repository Type Constants

Repository Type Constant	Value
MX_ADD_MEMBER_TASK	<process number for Provisioning>
MX_DEL_MEMBER_TASK	<process number for Deprovisioning>
MX_MODIFYTASK	<process number for Modify>
MX_OFFSET_ADD_MEMBER	<0>: On ValidFrom When to execute the Add member process
MX_REQ_PRIV	<Privilege reference (MSKEYVALUE)> Master privilege
REPOSITORY_SYNC	SYNC
REPOSITORY_TYPE	LDAP Repository type
INITIAL_LOAD	<Initial Load Job>
RESET_DELTA	<Reset Delta Job>

Related Information

[Identity Center: Repository Security](#)

9.2.1.10 Repository Constants for SAP HANA Database

Repository Constants

Repository Constant	Value
HANDLED_USER_PARAMS	< Editable user parameters on SAP HANA > Only parameters added to this constant can be edited or removed from the SAP HANA database. Default value: EMAIL ADDRESS, TIME ZONE, LOCALE Possible value: CLIENT (that is, <code>session client</code>) and custom user parameters If you leave the value empty, no editable user parameters will be provisioned to the SAP HANA database even though you have defined them in the Identity Management system. For more information about <code>session client</code>, see Setting Up and Provisioning Session Client and Custom User Parameters for SAP HANA Users [page 637]
IMDB_HOSTNAME	< Host name of the SAP HANA database server>
IMDB_PORT	< Port number of the SAP HANA database server> Usually 30015 + instance number * 100, for example, 32015 for instance number 20.>
IMDB_USER	< User ID> for authentication to the SAP HANA database
IMDB_PASSWORD	<password> for authentication to the SAP HANA database
IMDB_ADDITIONAL_JDBC_PARAMS	<empty> For experts: You may add special JDBC parameters here if there are no other repository constants that you can use for the required purpose.
IMDB_JDBC_DRIVER	Class name of the JDBC driver for the SAP HANA database. Usually the driver is delivered as part of the SAP HANA database with the <code>ngdbc.jar</code> file. Default value: com.sap.db.jdbc.Driver
IMDB_SSL_KEYSTORE	<empty> Keystore for private keys, used for X.509 client certificate authentication. Keystore name or path to keystore file. Default value: VM default.

Repository Constant	Value
IMDB_SSL_KEYSTORE_PASSWORD	<empty> Keystore password, related to IMDB_SSL_KEYSTORE. Default value: VM default
IMDB_SSL_KEYSTORE_TYPE	<empty> Keystore type, related to IMDB_SSL_KEYSTORE. Default value: VM default
IMDB_SSL_TRUSTSTORE	<empty> Keystore for trusted certificates, used for verification of server and CA certificates. Keystore name or path to key-store file. Default value: VM default
IMDB_SSL_TRUSTSTORE_PASSWORD	<empty> Keystore password, related to IMDB_SSL_TRUSTSTORE. Default value: VM default
IMDB_SSL_TRUSTSTORE_TYPE	<empty> Keystore type, related to IMDB_SSL_TRUSTSTORE. Default value: VM default
SYSTEM_PRIVILEGE	PRIV:SYSTEM:<Repository> <Repository> is the exact name of the repository to which the constants belong. In SAP Identity Management Administration User Interface, the value of the SYSTEM_PRIVILEGE repository constant is read only. When a repository is created, the name of the repository is automatically filled in.

Optional Repository Constants

Repository Constant	Value
IMDB_SSL_ENCRYPT	Switch for encryption of the connection to the SAP HANA database. Default: disabled

The following constants are repository type constants. They are valid for all repositories of this type and visible only in SAP Identity Management Developer Studio.

Repository Type Constants

Repository Type Constant	Value
MX_ADD_MEMBER_TASK	<process number for Provisioning>
MX_DEL_MEMBER_TASK	<process number for Deprovisioning>

Repository Type Constant	Value
MX_MODIFYTASK	<process number for Modify>
MX_OFFSET_ADD_MEMBER	<0>: On ValidFrom When to execute the Add member process
MX_REQ_PRIV_PCYADD_MISSING	<1>: Wait, that is wait for the privilege to be assigned Policy when the master privilege is missing
MX_REQ_PRIV_TIMEOUT	<0> Number of seconds to wait before timeout of the Wait policy
MX_REQ_PRIV	<Privilege reference (MSKEYVALUE)> Master privilege
REPOSITORY_SYNC	SYNC
REPOSITORY_TYPE	SAP_IN_MEMORY_DB Repository type
INITIAL_LOAD	<Initial Load Job>

9.2.1.11 Repository Constants for SuccessFactors

Repository Constants

Repository Constants	Value
CONNECT_TIMEOUT	Timeout interval for the connection to Success Factors
PROXY_HOST	Host for the HTTP proxy used to connect Success Factors
PROXY_PASSWORD	Password for the HTTP proxy used to connect Success Factors
PROXY_PORT	Port for the HTTP proxy used to connect Success Factors
PROXY_USER	User for the HTTP proxy used to connect Success Factors
READ_TIMEOUT	The time between sending request to Success Factors and receiving the response

Repository Constants	Value
AUTH_TYPE	The authentication method used. Possible values: <i>basic</i> - This is the default value. <i>mTLS</i> (mutual Transport Layer Security) Note The repository constant is case insensitive.
COMPANY_ID	Used if mTLS is configured as authentication method. Company ID of your SAP SuccessFactors system.
CLIENT_CERTIFICATE_PATH	Used if mTLS is configured as authentication method. File location (absolute path) that contains the client certificate along with its private key for connecting to a Success Factors system. For example: C:\MyFolder\Certificates\SFSF\SFCert.p12 or /MyFolder/Certificates/SFSF/SFCert.p12
CLIENT_CERTIFICATE_PASSWORD	Used if mTLS is configured as authentication method. Client certificate password used for connecting to your Success Factors system.
SFSF_HOST	The host name of the data center that hosts the SuccessFactors instance
SFSF_PASSWORD	Used if basic is configured as authentication method. Technical user's password
SFSF_PORT	Port number of the data center
SFSF_PROTOCOL	The attribute defines whether the data center is called via HTTP or HTTPS
SFSF_USER	Used if basic is configured as authentication method. User name and company ID of the technical user on SuccessFactors in the following format: username@companyID
SFSF_USE_EC_SOAP_API	Boolean constant describing whether EC SOAP API will be used
TRUSTSTORE	Name of the Java truststore that contains the trusted server certificate of the Success Factors instance. For example: C:\path\to\jre\lib\security\cacerts.

Repository Constants	Value
TRUSTSTORE_PASSWORD	Password for the truststore that will be used for connecting to Success Factors
SYSTEM_PRIVILEGE	PRIV:SYSTEM:<Repository> <Repository> is the exact name of the repository to which the constants belong. In SAP Identity Management Administration User Interface, the value of the SYSTEM_PRIVILEGE repository constant is read only. When a repository is created, the name of the repository is automatically filled in.

The following constants are repository type constants. They are valid for all repositories of this type and visible only in SAP Identity Management Developer Studio.

Repository Type Constants

Repository Type Constant	Value
MX_ADD_MEMBER_TASK	<process number for Provisioning>
MX_DEL_MEMBER_TASK	<process number for Deprovisioning>
REPOSITORY_SYNC	SYNC
INITIAL_LOAD	Initial Load Job
RESET_DELTA	Reset Delta Job

Related Information

[Setting Up a SuccessFactors System \[page 640\]](#)

9.2.1.12 Repository Constants for Lotus Notes

Repository Constants

Repository Constants	Usage	Value
REGISTRATION_SERVER_HOST-NAME	Mandatory	Registration Server of your Domino Domain

Repository Constants	Usage	Value
REGISTRATION_SERVER_USER-NAME	Mandatory	The user name for the Registration Server Enter the user name of the technical user that is used to establish the connection to the Domino server. Note The technical user needs at least Editor access level to the Names.nsf database. This access level allows the technical user to create and edit all documents in the database.
REGISTRATION_SERVER_PASS-WORD	Mandatory	The password for the Registration Server Enter the password of the technical user that is used to establish the connection to the Domino server.
REGISTRATION_SERVER_IIO_P_PORT	Mandatory	The IIO_P port of the Registration Server
REGISTRATION_SERVER_USE_SSL	Mandatory	Set up the SSL communication with the server. Default value: <i>false</i> Possible values: <i>true</i> (all requests use SSL) <i>false</i> (requests do not use SSL)
NAMES_DATABASE	Mandatory	The Lotus Notes Names database you provision to. Default value: <i>Names.nsf</i> This repository constant is used in the following operations: Create User, Create Group, Modify User, Modify Group, Assign Member and Unassign Member.
SYSTEM_PRIVILEGE	Mandatory Read-only	PRIV:SYSTEM:<Repository> <Repository> is the exact name of the repository to which the constants belong. In SAP Identity Management Administration User Interface, the value of the SYSTEM_PRIVILEGE repository constant is read only. When a repository is created, the name of the repository is automatically filled in.
USER_PREFIX	Mandatory	The Lotus Notes users created in SAP Identity Management will have an MSKEYVALUE= USER_PREFIX+NOTEID. Default value: <i>LN</i>

Repository Constants	Usage	Value
CERTIFICATE_AUTHORITY	Mandatory	The name of the certificate authority as defined on the Domino server. Default value: <i>Empty string</i> Possible values: • <i>Empty string</i> (no certificate authority is defined) • The name of the certificate authority Note If you add values for both CERTIFICATE_AUTHORITY and CERTIFIER_FILE repository constants, the value of the CERTIFIER_FILE repository constant is taken into account which results in requesting the certifier file, not the certificate authority. This repository constant is used in the following operations: Create User, Rename User, Recertify User, Delete User and Delete Group.
CERTIFIER_FILE	Mandatory	The certifier file must be on the Domino server. Enter the path to the certifier file on the Domino server. This repository constant is used in the following operations: Create User, Rename User, Recertify User, Delete User and Delete Group.
CERTIFIER_PASSWORD	Mandatory	Enter the certifier password for the certifier file. This repository constant is used in the following operations: Create User, Rename User, Recertify User, Delete User and Delete Group.
IS_IMMEDIATE	For specific operations only	Indicates if the deletion of all references to the user or to the group in the Domino Directory is immediate. Default value: <i>true</i> Possible values: • <i>true</i> (to delete all references to the user in the Domino Directory before issuing an administration process request) • <i>false</i> (to let the administration process make all deletions) This repository constant is used in the following operations: Delete User and Delete Group.

Repository Constants	Usage	Value
DENY_GROUP	For specific operations only	Enter the name of an existing group of type <i>Deny List Only</i> to which the name of the deleted user is added. Default value: "" Possible values: "" (The empty string means do not add the user name to any group) - Group name of type <i>Deny List Only</i> This repository constant is used in the following operations: Delete User.
MAIL_FILE_ACTION	For specific operations only	Indicates the disposition of the user's mail file. Default value: 2 Possible values: 0 (leaves the user's mail file) 1 (deletes the mail file on the user's home server) 2 (deletes the mail file on the user's home server and all replicas) This repository constant is used in the following operations: Delete User.
DESTINATION_SERVER_HOSTNAME	For specific operations only	Enter the hostname of the server (in canonical format) to which users' mail files are replicated after registration. This repository constant is used in the following operations: Create User.
PASSWORD_RESET_AGENT_DATA-BASE	For specific operations only	Enter the name of the Notes Storage Facility (NSF) database where the password reset agent is stored. For example: <i>passwordreset.nsf</i> This repository constant is used in the following operations: Reset Password.
PASSWORD_RESET_AGENT_NAME	For specific operations only	Enter the name of the password reset agent in the database. For example: <i>PasswordResetter</i> This repository constant is used in the following operations: Reset Password.

The following constants are repository type constants. They are valid for all repositories of this type and visible only in SAP Identity Management Developer Studio.

Repository Type Constants

Repository Type Constant	Value
MX_ADD_MEMBER_TASK	<process number for Provisioning>
MX_DEL_MEMBER_TASK	<process number for Deprovisioning>
MX_MODIFYTASK	<process number for Modify>
REPOSITORY_SYNC	SYNC
REPOSITORY_TYPE	NOTES Repository Type
INITIAL_LOAD	<Initial Load Job>

9.2.1.13 Repository Constants for Identity Authentication System

Repository Constants

Repository Constants	Value
SCI_HOST	Identity Authentication system hostname
SCI_PORT	Default: 443 Default port for https
SCI_USER	User name of the technical user
SCI_PASSWORD	Password of the technical user
PROXY_HOST	Proxy hostname
PROXY_PORT	Proxy port
PROXY_USER	User for proxy authentication
PROXY_PASSWORD	Password for proxy user
TRUSTSTORE	File location of the truststore to be used for establishing a secure connection
TRUSTSTORE_PASSWORD	Password to access the truststore (only needed when using certificate authentication).

Repository Constants	Value
READ_TIMEOUT	Default: 60000 Milliseconds to wait during read operations.
CONNECT_TIMEOUT	Default: 60000 Milliseconds to wait when making the connection.
CONNECTION_KEEPAIVE	Default: 60000 Milliseconds to wait before killing the connection
SYSTEM_PRIVILEGE	PRIV:SYSTEM:<Repository> <Repository> is the exact name of the repository to which the constants belong. In SAP Identity Management Administration User Interface, the value of the SYSTEM_PRIVILEGE repository constant is read only. When a repository is created, the name of the repository is automatically filled in.

The following constants are repository type constants. They are valid for all repositories of this type and visible only in SAP Identity Management Developer Studio.

Repository Type Constants

Repository Type Constant	Value
HTTP_PROTOCOL	https Protocol used for connection.
MX_ADD_MEMBER_TASK	<process number for Provisioning>
MX_DEL_MEMBER_TASK	<process number for Deprovisioning>
MX_MODIFYTASK	<process number for Modify>
REPOSITORY_SYNC	SYNC
REPOSITORY_TYPE	SCI
INITIAL_LOAD	<Initial Load Job>
MX_PRIV_GROUPING_ATTRIBUTE	
MX_PRIV_GROUPING_RULE	P:-1

9.2.1.14 Repository Constants for a SCIM System

General Usage

Repository Constants	Value
AUTH_TYPE	Authentication method to be used. Case insensitive. Possible values: basic and oauth Note In case of OAuth, SAP Identity Management is using basic authentication only the first time it authenticates with an authorization server. Basic authentication in both methods (basic and OAuth) is of type challenge-response.
AUTH_USER	Client ID of the created OAuth client Note This value is case sensitive.
AUTH_PASSWORD	Client secret of the created OAuth client
CONNECT_TIMEOUT	Milliseconds to wait when opening the connection Default: 60000
CONNECTION_KEEPALIVE	Milliseconds to wait before closing the connection Default: 60000
OAUTH_URL	OAuth 2.0 token service URL that is used to obtain a token.
PROXY_HOST	Proxy host name/IP address
PROXY_PASSWORD	Password for the proxy user
PROXY_PORT	Proxy port
PROXY_USER	User for the proxy authentication
READ_TIMEOUT	Milliseconds to wait during read operations Default: 60000

Repository Constants	Value
SCIM_HOST	Host name where the SCIM system for the customer's tenant is accessible
SCIM_PATH	Path after the SCIM host and port, specifying which SCIM system to use (example: <code>/<system>/api/v1/scim/12345abcde-a123-1234-aaaa-123456789</code>) Note Always put a slash ("/") at the beginning of the string, otherwise it will not generate a correct URL and your jobs will fail.
SCIM_PORT	443
SCIM_PROTOCOL	https
SYSTEM_PRIVILEGE	PRIV:SYSTEM:%\$rep.\$NAME%
TRUSTSTORE	File location of the Java truststore that contains the SCIM server's certificate, only needed when using HTTPS
TRUSTSTORE_PASSWORD	Password to access the truststore, only needed when using HTTPS
SCIM_ASSIGNMENT_METHOD	The HTTP method that will be executed when an assignment operation is triggered. For example: PUT or PATCH. Default value: PUT This is a repository type constant with override. It is visible and modifiable in SAP Identity Management Administration User Interface.
SCIM_MODIFY_METHOD	The HTTP method that will be executed when a modify operation is triggered. Default value: PUT This is a repository type constant. It is valid for all repositories of this repository type (SCIM) and visible only in SAP Identity Management Developer Studio.

SAP Cloud Identity Services - Identity Provisioning

Repository Constants	Value
AUTH_TYPE	Authentication method to be used. If your tenant is running on SAP BTP, Neo environment, enter oauth. If your tenant is running on SAP Cloud Identity Services infrastructure, enter basic. The value is case insensitive. Note If you import a .csv file generated from the Identity Provisioning UI, this value will be automatically entered.
AUTH_USER	If your tenant is running on SAP BTP, Neo environment, enter the OAuth client ID of the OAuth client. If your tenant is running on SAP Cloud Identity Services infrastructure, enter the Client ID of the technical user of type System created in the administration console of SAP Cloud Identity Services. For more information, see step 2 from the Hybrid Scenario procedure.
AUTH_PASSWORD	If your tenant is running on SAP BTP, Neo environment, enter the OAuth client secret. If your tenant is running on SAP Cloud Identity Services infrastructure, enter the client secret of the technical user of type System created in the administration console of SAP Cloud Identity Services. For more information, see step 2 from the Hybrid Scenario procedure.
CONNECT_TIMEOUT	Milliseconds to wait when opening the connection Default: 60000
CONNECTION_KEEPALIVE	Milliseconds to wait before closing the connection Default: 60000

Repository Constants	Value
OAUTH_URL	If your tenant is running on SAP BTP, Neo environment, enter the OAuth 2.0 service URL needed to obtain a token. If your tenant is running on SAP Cloud Identity Services infrastructure, remove the value of this constant. As the authentication type is basic, you do not need an OAuth URL to obtain a token. Note If you import a .csv file generated from the Identity Provisioning UI, this URL will be automatically entered.
PROXY_HOST	<your_proxy_host>
PROXY_PASSWORD	Password for the proxy user. It's automatically provided.
PROXY_PORT	<your_proxy_port>
PROXY_USER	You can skip this field.
READ_TIMEOUT	Milliseconds to wait during read operations Default: 60000
SCIM_HOST	For tenants running on SAP BTP, Neo environment, enter the proxy application URL, as you delete https:// and /ipsproxy. For tenants running on SAP Cloud Identity Services infrastructure, the value must point to the <ias-tenant-host> of your Identity Provisioning tenant. Note If you import a .csv file generated from the Identity Provisioning UI, this URL will be automatically entered in the proper format.

Repository Constants	Value
SCIM_PATH	For tenants running on SAP BTP, Neo environment, enter <code>/ipsproxy/api/v1/scim/<proxy_system_id></code>, where <code><proxy_system_id></code> is the ID of the proxy system. For tenants running on SAP Cloud Identity Services infrastructure, add service in the path, as follows: <code>/ipsproxy/service/api/v1/scim/<proxy_system_id></code> ⚠ Caution Always put a slash ("/") at the beginning of the string, otherwise it will not generate a correct URL and your jobs will fail. ℹ Note If you import a <code>.csv</code> file generated from the Identity Provisioning UI, this path will be automatically entered.
SCIM_PORT	443
SCIM_PROTOCOL	https
SYSTEM_PRIVILEGE	PRIV:SYSTEM:<repository_name> ℹ Note If you import a <code>.csv</code> file generated from the Identity Provisioning UI, this privilege will be automatically entered.
TRUSTSTORE	File location of the Java truststore that contains the SCIM server's certificate, only needed when using HTTPS
TRUSTSTORE_PASSWORD	Password to access the truststore, only needed when using HTTPS
SCIM_ASSIGNMENT_METHOD	HTTP method for an assignment operation, for example: PUT or PATCH. Default value: PUT
SCIM_MODIFY_METHOD	HTTP method for a modification operation Default value: PUT This is a repository type constant. It is valid for all repositories of this repository type (SCIM) and visible only in SAP Identity Management Developer Studio.

Related Information

[Setting up a SCIM System \[page 646\]](#)

9.2.1.15 Repository Constants for SAP S/4HANA

Repository Constants

Repository Constant	Value
S4HANA_HOSTNAME	Hostname or IP address of the SAP S/4HANA backend system.
S4HANA_PORT	Port number of the SAP S/4HANA backend system.
S4HANA_USERNAME	Username of the technical user for connecting to S/4HANA backend system.
S4HANA_PASSWORD	Password of the technical user for connecting to S/4HANA backend system.
IS_HTTPS	HTTP Protocol used for connection (http or https).
IS_HR_INTEGRATION_ACTIVE	Enable this setting only if SAP S/4HANA integration with HR system (such as, SAP HCM or SAP SuccessFactors) is active (that is, ON). In this case, S/4HANA connector will switch to READ-ONLY mode. This means the following: • You can only read data from S/4HANA system, that is, you can run the Initial Load job. • You cannot perform operations like create, update and delete of business users. These are not permitted and no requests are sent to S/4HANA backend system. • Users in SAP Identity Management will have the S/4HANA account and system privileges, but will not be created in the S/4HANA backend system.

Repository Constant	Value
READ_MARKED_FOR_ARCHIVING_USERS	Specify whether you want to read archived business users from SAP S/4HANA backend system. Possible values: • <Empty string> - Reads all business users (including marked for archiving and not marked for archiving business users). • true - Reads only marked for archiving business users. • false - Reads only active (not marked for archiving) business users. Marked for archiving business users will be skipped. Any value different from the values listed above will result in fail of the Initial Load job.
CONNECTION_TIMEOUT	The timeout interval for creating, modifying or deleting business users in SAP S/4HANA backend system in milliseconds. If no value is provided or the provided value is invalid, then a default value will be used: 1000000 (15 minutes).
READ_TIMEOUT	The timeout interval for reading business users from SAP S/4HANA backend system in milliseconds. If no value is provided or the provided value is invalid, then a default value will be used: 12000000 (3 hours and 20 minutes).
SAP_CLIENT_NUMBER	ABAP system ID Clients are identified with a three-digit number. Any value different from number will result in fail of the Initial Load job.
SOAP_SERVICE_PREFIX	Web service path From the S/4HANA SOAP service URL, enter the part of the URL between https://<host><port> and <service name>. For example, given the following URL: https://<host>:<port>/sap/bc/srt/scs/sap/<service name>, you need to enter: /sap/bc/srt/scs/sap/ The value may start and end with "/" (backslash). If it's missing, it will be automatically appended because it is used as SOAP URL delimiter.

Repository Constant	Value
SYSTEM_PRIVILEGE	PRIV:SYSTEM:<Repository> <Repository> is the exact name of the repository to which the constants belong. In SAP Identity Management Administration User Interface, the value of the SYSTEM_PRIVILEGE repository constant is read only. When a repository is created, the name of the repository is automatically filled in.

The following constants are repository type constants. They are valid for all repositories of this type and visible only in SAP Identity Management Developer Studio.

Repository Type Constants

Repository Type Constant	Value
MX_ADD_MEMBER_TASK	<process number for Provisioning>
MX_DEL_MEMBER_TASK	<process number for Deprovisioning>
MX_MODIFYTASK	<process number for Modify>
MX_REQ_PRIV	<Privilege reference (MSKEYVALUE)> Master privilege
MX_PRIV_GROUPING_RULE	P:-1
MX_PRIV_GROUPING_ATTRIBUTE	
REPOSITORY_TYPE	S4HANA
REPOSITORY_SYNC	SYNC
INITIAL_LOAD	<Initial Load Job>

9.2.2 Attribute Mappings

This section describes the attributes that are supported by the connectors in the SAP provisioning framework and how they are mapped to attributes in the Identity Management.

Related Information

[Mapping Between Identity Management and AS ABAP Attributes \[page 1277\]](#)

[Additional Mapping Attributes for Enhanced SAP Business Suite Integration \[page 1289\]](#)

[Mapping Between Identity Management and AS Java Attributes \[page 1293\]](#)

[Mapping Between Identity Management and HANA Attributes \[page 1295\]](#)

[Mapping Between Identity Management and SuccessFactors Attributes \[page 1300\]](#)

[Mapping Between Identity Management and Domino Technical Attributes \[page 1301\]](#)

[Mapping Between Identity Management and Active Directory Attributes \[page 1304\]](#)

9.2.2.1 Mapping Between Identity Management and AS ABAP Attributes

The following table shows the ABAP attributes that are supported by the ABAP connector and how they are mapped to attributes in the Identity Management.

Note

Attributes that are new to Release 7.1 or 7.2 are indicated accordingly.

For more information about the Business Add-Ins (BAdIs) used by SAP Identity Management, see [Extending the SAP Provisioning Framework](#)

Attribute Mapping

SAP Identity Management Attribute	Java (SPML) Attribute	ABAP Connector Attribute	BAPI Parameter	BAPI Field	HR Field	LDAP Mapping to InetOrgPerson
DISPLAYNAME	displayname	Displayname	ADDRESS	FULLNAME	P0001-ENAME	cn
MSKEYVALUE	logonname	logonuid	USERNAME		SYHR_A_P0105_AF_SYS UN-AME	
MX_ACA-DEMIC_TITLE_1		AddressTitleAca1	ADDRESS	TITLE_ACA1	TEXT_P0002-TITEL	title
MX_ACA-DEMIC_TITLE_2		AddressTitleAca2	ADDRESS	TITLE_ACA2		
MX_ACCESSIBILITYLEVEL		WebAccessibility				
MX_ACCOUNTING_NUMBER		LogondataAccnt	LOGONDATA	ACCNT		

SAP Identity Management Attribute	Java (SPML) Attribute	ABAP Connector Attribute	BAPI Parameter	BAPI Field	HR Field	LDAP Mapping to InetOrgPerson
MX_AD-DRESS_BUILD-ING		AddressBuild-Long	ADDRESS	BUILD_LONG		
MX_AD-DRESS_CHECK STATUS			ADDRESS	CHCKSTATUS		
MX_AD-DRESS_CITY			ADDRESS	CITY	WORKCEN-TER_CITY	I
MX_AD-DRESS_CITY_NO			ADDRESS	CITY_NO		
MX_AD-DRESS_CO_NAME		AddressCO-Name	ADDRESS	C_O_NAME		
MX_AD-DRESS_COMPANY_POSTAL_CODE		AddressPostl-Cod3	ADDRESS	POSTL_COD3	WORKCEN-TER_POSTCODE	
MX_AD-DRESS_COUNTRY		AddressCountry	ADDRESS	COUNTRY	WORKCEN-TER_COUNTRY	
MX_AD-DRESS_DIFFERENT_CITY		AddressHomeCity	ADDRESS	HOME_CITY		
MX_AD-DRESS_DIFFERENT_CITY_NO		AddressHome-cityno	ADDRESS	HEMECITYNO		
MX_AD-DRESS_DISTRICT		AddressDistrict	ADDRESS	DISTRICT		
MX_AD-DRESS_DISTRICT_NO		AddressDistrictNo	ADDRESS	DISTRICT_NO		

SAP Identity Management Attribute	Java (SPML) Attribute	ABAP Connector Attribute	BAPI Parameter	BAPI Field	HR Field	LDAP Mapping to InetOrgPerson
MX_AD-DRESS_FLOOR		AddressFloor	ADDRESS	FLOOR		
MX_AD-DRESS_HOUSE_NO		AddressHouseNo	ADDRESS	HOUSE_NO		
MX_AD-DRESS_HOUSE_NO_SUPPLEMENT		AddressHouseNo2	ADDRESS	HOUSE_NO2		
MX_AD-DRESS_LAN-GUAGE		AddressLanguageISO	ADDRESS	LANGUIISO		
MX_AD-DRESS_NAME_1		AddressName	ADDRESS	NAME		
MX_AD-DRESS_NAME_2		AddressName2	ADDRESS	NAME_2		
MX_AD-DRESS_NAME_3		AddressName3	ADDRESS	NAME_3		
MX_AD-DRESS_NAME_4		AddressName4	ADDRESS	NAME_4		
MX_AD-DRESS_NOTES		AddressAddressNotes	ADDRESS	ADR_NOTES		
MX_AD-DRESS_POBOX		AddressPoBox	ADDRESS	PO_BOX		postofficebox
MX_AD-DRESS_PO-BOX_CITY		AddressPoBox-City	ADDRESS	PO_BOX_CITY		
MX_AD-DRESS_PO-BOX_COUNTRY		AddressPoBoxCountry	ADDRESS	POBOX_CTRY		

SAP Identity Management Attribute	Java (SPML) Attribute	ABAP Connector Attribute	BAPI Parameter	BAPI Field	HR Field	LDAP Mapping to InetOrgPerson
MX_AD-DRESS_PO-BOX_POSTAL_CODE		AddressPostl-Cod2	ADDRESS	POSTL_COD2		
MX_AD-DRESS_PO-BOX_REGION		AddressPoBox-Reg	ADDRESS	REGION		
MX_AD-DRESS_PO-BOX_WITHOUT_NUMBER		AddressPo-WONo	ADDRESS	PO_W_O_NO		
MX_AD-DRESS_POSTAL_CODE		AddressPostl-Cod1	ADDRESS		postalcode	
MX_AD-DRESS_REASON_DONT_USE_POBOX_ADDRESS		AddressDontUseP	ADDRESS	DONT_USE_P		
MX_AD-DRESS_REASON_DONT_USE_STREET_ADDRESS		AddressDontUseS	ADDRESS	DONT_USE_S		
MX_AD-DRESS_REGION		AddressRegion	ADDRESS	REGION		
MX_AD-DRESS_REGION_GROUP		AddressRegiongroup	ADDRESS	REGIOGROUP		
MX_AD-DRESS_ROOM_NO		Address-RoomNo	ADDRESS	ROOM_NO		
MX_AD-DRESS_STREET_1		AddressStreet	ADDRESS	STREET		

SAP Identity Management Attribute	Java (SPML) Attribute	ABAP Connector Attribute	BAPI Parameter	BAPI Field	HR Field	LDAP Mapping to InetOrgPerson
MX_AD-DRESS_STREET_2		AddressStr-Suppl1	ADDRESS	STR_SUPPL1		
MX_AD-DRESS_STREET_3		AddressStr-Suppl2	ADDRESS	STR_SUPPL2		
MX_AD-DRESS_STREET_4		AddressStr-Suppl3	ADDRESS	STR_SUPPL3		
MX_AD-DRESS_STREET_5		AddressLocation	ADDRESS	LOCATION		
MX_AD-DRESS_STREET_NO		Address-StreetNo	ADDRESS	STREET_NO		
MX_AD-DRESS_STREET-ADDRESS					WORKCENTER_STREET	street
MX_AD-DRESS_TAX_JURISDICTION_CODE		AddressTaxjur-code	ADDRESS	TAXJURCODE		
MX_AD-DRESS_TIME_ZONE		AddressTime-Zone	ADDRESS	TIME_ZONE		
MX_AD-DRESS_TITLE		AddressTitle	ADDRESS	TITLE		
MX_AD-DRESS_TRANSPORT_ZONE		AddressTransp-zone	ADDRESS	TRANSPZONE		
MX_AD-MIN_UNIT		companyid	LOGONDATA	CLASS		
MX_BIRTH-NAME		AddressBirth-Name	ADDRESS	BIRTH_NAME		

SAP Identity Management Attribute	Java (SPML) Attribute	ABAP Connector Attribute	BAPI Parameter	BAPI Field	HR Field	LDAP Mapping to InetOrgPerson
MX_CATT_TES T_STATUS		DefaultsCatt- kennz	DEFAULTS	CATTKENNZ		
MX_CERTIFI- CATE						usercertificate
MX_COMMUNI- CATION_LAN- GUAGE		AddressLangu- PISO	ADDRESS	LANGUP_ISO		
MX_COMMUNI- CAT- ION_METHOD		AddressComm- Type	ADDRESS	COMM_TYPE		
MX_COSTCEN- TER		DefaultCost- center	DEFAULTS	KOSTL		
MX_DATEFOR- MAT		dateformat	DEFAULTS	DATFM		
MX_DEPART- MENT		department	ADDRESS	DEPARTMENT	P0001_OR- GEH_TL	ou
MX_EN- CRYPTED_PAS SWORD (7.2)		password				
MX_FAX_ADDI- TIONAL		additionalFaxes	ADDFAX			
MX_FAX_PRI- MARY	fax	primaryFax	ADDFAX		SYHR_A_P0105 _AF_FAX	facsimiletele- phonenummer
MX_FIRST- NAME	firstname	firstname	ADDRESS	FIRSTNAME	P0002-VORNA	givenname
MX_HCM_SY- SUNAME (7.1)					SYHR_A_P0105 _AF_SYS UN- AME	
MX_IDENTI- TYUUID (7.2)		identityuuid	IDENTITY	IDEN- TITY_UUID		

SAP Identity Management Attribute	Java (SPML) Attribute	ABAP Connector Attribute	BAPI Parameter	BAPI Field	HR Field	LDAP Mapping to InetOrgPerson
MX_IN-HOUSE_MAIL		AddressInhouseMI	ADDRESS	INHOUSE_ML		
MX_INITIALS		AddressInitials	ADDRESS	INITIALS		initials
MX_LANGUAGE		locale	DEFAULTS	LANGU	P0002-SPRSL	preferredlanguage
MX_LAST-NAME		lastname	ADDRESS	LASTNAME	P0002-NACHN	sn
MX_LASTMOD-TIME (7.1 SP 5)			LASTMODIFIED	MODDATE		
MX_LASTMODIFIER (7.1 SP 5)			LASTMODIFIED	MODIFIER		
To be defined		UClassBname-Chargeable	UCLASS	BNAME_CHAR GEABLE		
To be defined		UClassClient	UCLASS	CLIENT		
To be defined		UClassCountry-Surcharge	UCLASS	COUN- TRY_SUR- CHARGE		
To be defined		UClassLicType	UCLASS	LIC_TYPE		
To be defined		UClassSpecVers	UCLASS	SPEC_VERS		
To be defined		UClassSubstituteFrom	UCLASS	SUBSTI- TUTE_FROM		
To be defined		UClassSubstituteUntil	UCLASS	SYSID		
To be defined		UClassSysid	UCLASS	SYSID		
To be defined			UCLASSSYS	RCVSYSTEM		
To be defined		UClasssysBnameChargeable	UCLASSSYS	BNAME_CHAR GEABLE		

SAP Identity Management Attribute	Java (SPML) Attribute	ABAP Connector Attribute	BAPI Parameter	BAPI Field	HR Field	LDAP Mapping to InetOrgPerson
To be defined		UClasssysClient	UCLASSSYS	CLIENT		
To be defined		UClasssysCountrySurcharge	UCLASSSYS	COUNTRY_SURCHARGE		
To be defined		UClasssysLicType	UCLASSSYS	LIC_TYPE		
To be defined		UClasssysSpecVers	UCLASSSYS	SPEC_VERS		
To be defined		UClasssysSubstituteFrom	UCLASSSYS	SUBSTITUTE_FROM		
To be defined		UClasssysSubstituteUntil	UCLASSSYS	SUBSTITUTE_UNTIL		
To be defined		UClasssysSysid	UCLASSSYS	SYSID		
MX_LOCKED	islocked	islocked	ISLOCKED	LOCAL_LOCK		
MX_LOGONALIAS		useralias	ALIAS	USERALIAS		
MX_MAIL_ADDITIONAL		additionalMails	ADDSMTP		SYHR_A_P0105_AF_EMAIL	
MX_MAIL_PRIMARY	email	primaryMail	ADDSMTP		SYHR_A_P0105 mail_AF_EMAIL	
MX_MIDDLE-NAME		AddressMiddle-name	ADDRESS	MIDDLENAME	P0002-MIDNM	
MX_MOBILE_ADDITIONAL		additionalMobiles	ADDTEL			
MX_MOBILE_PRIMARY	mobile	primaryMobile	ADDTEL		SYHR_A_P0105 mobile_AF_CELL	
MX_NAME-COUNTRY (7.1)		AddressName-Country	ADDRESS	NAMCOUNTRY		
MX_NAME-ABBREVIATION		AddressInitsSig	ADDRESS	NITS_SIG		

SAP Identity Management Attribute	Java (SPML) Attribute	ABAP Connector Attribute	BAPI Parameter	BAPI Field	HR Field	LDAP Mapping to InetOrgPerson
MX_NAME_PREFIX_1		AddressPrefix1	ADDRESS	PREFIX1		
MX_NAME_PREFIX_2		AddressPrefix2	ADDRESS	PREFIX2		
MX_NAMEFORMAT (7.1)			ADDRESS	NAMEFORMAT		
MX_NICKNAME		AddressNick-name	ADDRESS	NICKNAME		
MX_NUMBERFORMAT		Numberformat	DEFAULTS	DCPFM		
MX_PAGER_ADDITIONAL		additionalPag-ers	ADDRESS			pager
MX_PAGER_PRIMARY		primaryPager	ADDPAG			
MX_PARAMETER (7.1)			PARAMETER1	PARID, PARVA		
MX_PASSWORD	password	password	PASSWORD	BAPIPWD		userpassword
MX_PASSWORD_DISABLED	ispassworddis-abled	Ispassworddis-abled	ISLOCKED	NO_USER_PW		
Mapping only applies to Initial Load and other usages of FromSAP/FromSAPI-identity passes (BAPI_USER_GET_LIST).						
MX_PERSON-UUID (7.2)		personuuid	IDENTITY	BPPERSON		
MX_PHONE_ADDITIONAL		additional-Phones	ADDTel			homephone

SAP Identity Management Attribute	Java (SPML) Attribute	ABAP Connector Attribute	BAPI Parameter	BAPI Field	HR Field	LDAP Mapping to InetOrgPerson
MX_PHONE_PRIMARY	telephone	primaryPhone	ADDTEL		SYHR_A_P0105_AF_TEL_NR + SYHR_A_P0105_AF_TEL_EXT	telephonenumber
MX_PRINTER-SETTINGS_SPDA		DefaultsSpda	DEFAULTS	SPDA		
MX_PRINTER-SETTINGS_SPDB		DefaultsSpdb	DEFAULTS	SPDB		
MX_PRINTER-SETTINGS_SPLD		DefaultsSpld	DEFAULTS	SPLD		
MX_PRINTER-SETTINGS_SPLG		DefaultsSplg	DEFAULTS	SPLG		
MX_PRT_ADDITIONAL (7.2)		additionalPRT	ADDPRT			
MX_REFERENCE_USER		ReferenceUser	REF_USER	REF_USER		
MX_RML_ADDITIONAL (7.1 SP 5)			ADDRML			
MX_RML_PRIMARY (7.2)		primaryRML	ADDRML			
MX_SALUTATION		salutation	ADDRESS	TITLE_P	T522T-ANRLT	
MX_SEARCH_TERM_1		AddressSort1P	ADDRESS	SORT1_P		
MX_SEARCH_TERM_2		AddressSort2P	ADDRESS	SORT2_P		

SAP Identity Management Attribute	Java (SPML) Attribute	ABAP Connector Attribute	BAPI Parameter	BAPI Field	HR Field	LDAP Mapping to InetOrgPerson
MX_SECOND-NAME		AddressSe- condname	ADDRESS	SECONDNAME		
MX_SECUR- ITY_POLICY		abapsecurity- policy	LOGONDATA	SECUR- ITY_POLICY		
MX_SNC_FLAG		SNCFlag	SNC	GUIFLAG		
MX_SNC_NAM E		SNCName	SNC	PNAME		
MX_SSF_ADDI- TIONAL (7.2)		additionISSF	ADDSSF			
MX_SSF_PRI- MARY (7.2)		primarySSF		ADDSSF		
MX_START_ME NU (7.1)		DefaultsStart- Menu	DEFAULTS	START_MENU		
MX_TIMEFOR- MAT (7.1 SP 5)			DEFAULTS	TIMEFM		
MX_TIMEFOR- MAT (7.2)		timeformat	DEFAULTS	TIMEFM		
MX_TIMEZONE	timezone	timezone	LOGONDATA	TZONE		
MX_TI- TLE_SUPPLE- MENT		AddressTi- tleSppl	ADDRESS	TITLE_SPPL		
MX_TLX_ADDI- TIONAL (7.2)		additionalTLX	ADDTLZ			
MX_TLX_PRI- MARY (7.2)		primaryTLX	ADDTLX			
MX_TTX_PRI- MARY (7.2)		primaryTTX	ADDTTX			
MX_TTX_ADDI- TIONAL (7.2)		additionalTTX	ADDTTX			

SAP Identity Management Attribute	Java (SPML) Attribute	ABAP Connector Attribute	BAPI Parameter	BAPI Field	HR Field	LDAP Mapping to InetOrgPerson
MX_URI_ADDITIONAL (7.1 SP5)		additionalURI	ADDURI			
MX_URI_PRIMARY (7.2)		primaryURI	ADDURI			
MX_USER_CATEGORY (7.1)		groups	GROUPS	USERGROUP		
MX_USERTYPE		securitypolicy	LOGONDATA	USTYP		
MX_VALIDFROM	validfrom	validfrom	LOGONDATA	GLTGV		
MX_VALIDTO	validto	validto	LOGONDATA	GLTGB		
MX_WORKPLACE_BUILDING		AddressBuildingP	ADDRESS	BUILDING_P	TEXT_P8001_BUILD	
MX_WORKPLACE_FLOOR		AddressFloorP	ADDRESS	FLOOR_P		
MX_WORKPLACE_FUNCTION		jobfunction	ADDRESS	FUNCTION	P0001_PLANS_TL	
MX_WORKPLACE_ROOM		AddressRoomNoP	ADDRESS	ROOM_NO_P	WORKCENTER_ROOM	
MX_X400_ADDITIONAL (7.2)		additionalX400	ADDX400			
MX_X400_PRIMARY (7.2)		primaryX400	ADDX400			
MX_USER_UID		userUuid	SAP-USER_UUID	SAP_UID		
MXREF_MX_COMPANY_ADDRESS		Company	COMPANY	COMPANY		
MXREF_MX_PRIVILEGE		roles	ACTIVITYGROUPS	AGR_NAME		

SAP Identity Management Attribute	Java (SPML) Attribute	ABAP Connector Attribute	BAPI Parameter	BAPI Field	HR Field	LDAP Mapping to InetOrgPerson
MXREF_MX_PR IVILEGE		profiles	profiles	BAPIPROF		

Related Information

[Setting Up an AS ABAP System \[page 623\]](#)

9.2.2.2 Additional Mapping Attributes for Enhanced SAP Business Suite Integration

The following table shows additional attributes that are used for the enhanced SAP Business Suite integration use case.

These attributes are used as parameters for the Business Add-In (BAI) implementations that apply to the specific integration scenario, for example, Customer Relationship Management (CRM). For more information about the BAIs used by SAP Identity Management, see [Extending the SAP Provisioning Framework](#)

Attributes that are explicitly used for the SAP Business Suite applications are indicated as such in the Identity Center schema using the prefix MX_FS so that they can be recognized and provisioned accordingly. For example, the SAP HCM attribute PERNR is mapped to MX_FS_PERSONNEL_NUMBER in the Identity Center schema.

Additional Attributes

IC Attribute	Short Description	Source System	Source Attribute Name (for example, DDIC Data Element)	BAI Parameter Name
DISPLAYNAME	Formatted Name of Employee or Applicant	SAP HCM	P0001-ENAME	ADDRESS-FULLNAME
MX_ACADEMIC_TITLE_1	Academic Title Text	SAP HCM	TEXT_P0002_TITLE	ADDRESS-TITLE_ACA1
MX_ADDRESS-STREETADDRESS	Street	SAP HCM	P8001-STRAS	ADDRESS-STREET (AD_STREET)
MX_ADDRESS_CITY	City	SAP HCM	P8001-ORT01	ADDRESS-CITY (AD_CITY1)

IC Attribute	Short Description	Source System	Source Attribute Name (for example, DDIC Data Element)	BAdI Parameter Name
MX_ADDRESS_COUN-TRY	Country	SAP HCM		ADDRESS-COUN-TRYISO
MX_FIRSTNAME	First Name	SAP HCM	P0002-VORN	ADDRESS-FIRST-NAME
MX_FS__POSITION_ID	Position	SAP HCM	P0001-PLANS	POSITION_ID
MX_FS_ACA-DEMIC_TITLE_1_ID	Academic Title	SAP HCM	P0002-TITLE	ACADEMIC_TI-TLE_1_ID
MX_FS_BP_PER-SON_ID	Business Partner (Per-son)		SYSUUID	BP_PERSON_ID
MX_FS_BUSI-NESS_AREA	Business Area Text	SAP HCM	TEXT_P0001_GSBER	BUSINESS_AREA
MX_FS_BUSI-NESS_AREA_ID	Business Area	SAP HCM	P0001-GSBER	BUSINESS_AREA_ID
MX_FS_CENTRALPER-SON_ID	Central Person (CP)	SAP HCM	OBJID (char 10)	CENTRALPERSON_ID
MX_FS_COM-PANY_CODE	Company Code Text	SAP HCM	TEXT_P0001-BUKRS	COMPANY_CODE
MX_FS_COM-PANY_CODE_ID	Company Code	SAP HCM	P0001-BUKRS	COMPANY_CODE_ID
MX_FS_COST_CEN-TER	Text Cost Center	SAP HCM	TEXT_P0001-KOSTL	COST_CENTER
MX_FS_COST_CEN-TER_ID	Cost Center	SAP HCM / SAP NetWeaver	P0001-KOSTL	COST_CENTER_ID
MX_FS_EM-PLOYEE_GROUP	Employee Group Text	SAP HCM	TEXT_P0001_PERSG	EMPLOYEE_GROUP
MX_FS_EM-PLOYEE_GROUP_ID	Employee Group	SAP HCM	P0001-PERSG	EM-PLOYEE_GROUP_ID
MX_FS_EM-PLOYEE_SUBGROUP	Employee Sub-group_Text	SAP HCM	TEXT_P0001_PERSK	EMPLOYEE_SUB-GROUP
MX_FS_EM-PLOYEE_SUBGROUP_ID	Employee Subgroup	SAP HCM	P0001-PERSK	EMPLOYEE_SUB-GROUP_ID

IC Attribute	Short Description	Source System	Source Attribute Name (for example, DDIC Data Element)	BAdI Parameter Name
MX_FS_EMPLOYMENT_STATUS_ID	Employment Status	SAP HCM	P0000-STAT2	EMPLOYMENT_STATUS_ID
MX_FS_EMPLOYMENT_STATUS	Employment Status Text	SAP HCM	TEXT_P0000_STAT2	EMPLOYMENT_STATUS
MX_FS_IDENTITY_TYPE	Type of Identity			IDENTITY_TYPE
MX_FS_JOB	Job Text	SAP HCM	P0001-STELL_TL	JOB
MX_FS_JOB_ID	Job	SAP HCM	P0001-STELL	JOB_ID
MX_FS_ORGANIZATIONAL_UNIT	Organizational Unit Text	SAP HCM	P0001-ORGEH_TL	ORGANIZATIONAL_UNIT
MX_FS_ORGANIZATIONAL_UNIT_ID	Organizational Unit	SAP HCM	P0001-ORGEH	ORGANIZATIONAL_UNIT_ID
MX_FS_PERNR_IS_MANAGER	PERNR_IS_MANAGER	SAP HCM		PERNR_IS_MANAGER
MX_FS_PERSONNEL_AREA	Personnel Area Text	SAP HCM	TEXT_P0001_WERKS	PERSONNEL_AREA
MX_FS_PERSONNEL_AREA_ID	Personnel Area	SAP HCM	P0001-WERKS	PERSONNEL_AREA_ID
MX_FS_PERSONNEL_NUMBER	Personnel number	SAP HCM	PERNR	PERSONNEL_NUMBER
MX_FS_PERSONNEL_NUMBER_OF_MANAGER	Personnel Number of nextlevel manager	SAP HCM	PERNR_OF_MANAGER	PERSONNEL_NUMBER_OF_MANAGER
MX_FS_PERSONNEL_SUBAREA	Personnel Subarea Text	SAP HCM	TEXT_P0001_BTRTL	PERSONNEL_SUBAREA
MX_FS_PERSONNEL_SUBAREA_ID	Personnel Subarea	SAP HCM	P0001-BTRTL	PERSONNEL_SUBAREA_ID
MX_FS_POSITION	Position Text	SAP HCM	P0001-PLANS_TL	POSITION
MX_FS_POSITION_ID	Position	SAP HCM	P0001-PLANS	

(7.2)

IC Attribute	Short Description	Source System	Source Attribute Name (for example, DDIC Data Element)	BAdI Parameter Name
MX_FS_SALUTATION_ID	Form-of-Address Key	SAP HCM	P0002-ANRED	SALUTATION_ID
MX_FS_SCMEWM_PRR_ID	EWM Processor	EWM	/SCMB/DE_PRR	SCMEWM_PRR_ID
MX_FS_SCMSNC_BP_ORG_ID	SNC Business Partner (Organization)	SCM, SNC	SYSUUID	SCMSNC_BP_ORG_ID
MX_FS_SCMSNC_VISIBILITY_PROFILE	SNC Visibility Profile	IDM		SCMSNC_VISIBILITY_PROFILE
MX_FS_SCMTMS_BP_ORG_ID	TM TSP (Organization Business Partner)	SCM TM	SYSUUID or BP Number	SCMTMS_BP_ORG_ID
MX_FS_SOURCE_SYSTEM	SourceSystem	SAP NetWeaver	SY-SYSID+ SY-MANDT	SOURCE_SYSTEM
MX_FS_WORK_CONTRACT	Contract Text	SAP HCM	TEXT_P0001-ANSVH	WORK_CONTRACT
MX_FS_WORK_CONTRACT_ID	Contract (in CE mode, should be renamed to Personnel Assignment)	SAP HCM	P0001-ANSVH	WORK_CONTRACT_ID
MX_HCM_SYSUSERNAME	System user	SAP HCM	P0105_AF-SYSUSERNAME	
MX_LANGUAGE	Language	SAP HCM	P0002-SPRSL	DEFAULTS-LANGU
MX_LASTNAME	Last Name	SAP HCM	P0002-NACHN	ADDRESS-LASTNAME
MX_MAIL_PRIMARY or MX_MAIL_ADDITIONAL	E-Mail	SAP HCM or IDM	P0105_AF-EMAIL	ADDRESS-E_MAIL or table ADDSMTP
MX_MOBILE_PRIMARY	Cell Phone	SAP HCM	P0105_AF-CELL	List of numbers in table ADDTEL

IC Attribute	Short Description	Source System	Source Attribute Name (for example, DDIC Data Element)	BAdI Parameter Name
MX_PHONE_PRIMARY or MX_PHONE_ADDITIONAL	Phone	SAP HCM	P0105_AF-TEL_NR + P0105_AF-TEL_EXT	ADDRESS-TEL1_NUMBR (First telephone no.: dialing code+number) ADDRESS-TEL1_EXT (First Telephone No.: Extension) or list of numbers in table ADD-TEL
MX_WORK-PLACE_ROOM	Room Number	SAP HCM	P1028-ROOM1 (P8001-ROOM1)	ADDRESS-ROOM_NO_P
MX_SALUTATION	Form-of-Address Text	SAP HCM	TEXT_P0002_ANRED	ADDRESS-TITLE_P
MX_WORK-PLACE_BUILDING	Building Number	SAP HCM	P1028-BUILDING (P8001-BUILDING)	ADDRESS-BUILDING_P
MX_USER_UUID	User UUID generated by Identity Authentication service	Identity Authentication Service		SAPUSER_UUID-SAP_UID

9.2.2.3 Mapping Between Identity Management and AS Java Attributes

The following tables show the AS Java attributes that are supported by the AS Java connector and how they are mapped to attributes in the Identity Management.

User Attribute Mappings

SAP Identity Management Attribute	AS Java Attribute	UME Data Source (LDAP Directory or DB of SAP NetWeaver AS Java)
MSKEYVALUE	SPMLID	DB, LDAP
DISPLAYNAME	displayname	LDAP
ACCOUNT<Repository>	SPMLID	DB, LDAP
MX_FAX_ALL	fax	LDAP
MX_FAX_PRIMARY	fax	DB, LDAP

SAP Identity Management Attribute	AS Java Attribute	UME Data Source (LDAP Directory or DB of SAP NetWeaver AS Java)
MX_FIRSTNAME	firstname	DB, LDAP
MX_LASTNAME	lastname	DB, LDAP
MX_MAIL_PRIMARY	email	DB, LDAP
MX_MOBILE_PRIMARY	mobile	DB, LDAP
MX_LOCKED	islocked	DB, LDAP
MX_PASSWORD_DISABLED	ispassworddisabled	DB, LDAP
MX_LANGUAGE	locale	DB, LDAP
MX_PASSWORD		DB, LDAP
MX_ENCRYPTED_PASSWORD		DB, LDAP
MX_USERTYPE	securitypolicy	LDAP
MX_LANGUAGE_COUNTRY	locale	DB, LDAP
MX_LANGUAGE_VARIANT	locale	DB, LDAP
MX_CERTIFICATE	cartificate	DB, LDAP
MX_ADMIN_UNIT	companyid	LDAP
MX_ADDRESS_CITY	city	LDAP
MX_ADDRESS_COUNTRY	country	DB, LDAP
MX_ADDRESS_POBOX	pobox	DB, LDAP
MX_ADDRESS_POSTAL_CODE	zip	DB, LDAP
MX_ADDRESS_REGION	state	DB, LDAP
MX_DEPARTMENT	department	DB, LDAP
MX_WORKPLACE_FUNCTION	position	DB, LDAP
MX_PHONE_PRIMARY	telephone	DB, LDAP
MX_TIMEZONE	timezone	DB, LDAP
MX_VALIDFROM	validfrom	DB, LDAP
MX_VALIDTO	validto	DB, LDAP

SAP Identity Management Attribute	AS Java Attribute	UME Data Source (LDAP Directory or DB of SAP NetWeaver AS Java)
MX_ACCESSIBILITYLEVEL	accessibilitylevel	DB

Group Attribute Mappings

Identity Center Attribute	AS Java Attribute	UME Data Source is LDAP Directory or DB of SAP NetWeaver AS Java
MSKEYVALUE	SPMLID	DB, LDAP
ACCOUNT<Repository>	SPMLID	DB, LDAP
DISPLAYNAME	displayname	DB, LDAP
MX_REPOSITORYNAME		DB, LDAP
DESCRIPTION	description	DB, LDAP

Related Information

[Setting Up an AS Java System \[page 628\]](#)

9.2.2.4 Mapping Between Identity Management and HANA Attributes

The following table shows the HANA attributes that are supported by the SAP HANA connector and how they are mapped to attributes in the Identity Management.

Attribute Mappings

HANA Attribute	SAP Identity Management Attribute	Mapping Format	Description
USER_NAME	ACCOUNT<Repository>/MSKEYVALUE	String	For SAP HANA systems, the default value provisioned as USER_NAME is the Account<Repository> / MSKEYVALUE of the entry. However, due to limitations of permitted characters in In-Memory database user names, you might want to implement your custom logic to generate a user name for your In-Memory database users here. This customizations should be implemented in the script sap_hana_generateUsername.
EXTERNAL_IDENTITY	MX_KERBEROS_IDENTITY	String	This attribute contains Kerberos identity and is used for Kerberos authentication.
VALID_FROM	MX_VALIDFROM	Date or Date and Time	This attribute shows the start date of user's validity.
VALID_UNTIL	MX_VALIDTO	Date or Date and Time	This attribute shows the end date of user's validity.
USER_DEACTIVATED	MX_DISABLED	Boolean	The attribute is true if the user is disabled.
EMAIL ADDRESS	MX_MAIL_PRIMARY	String	This attribute shows the email address of the user.
 Note Please have in mind that this attribute is unique in SAP HANA systems – so that you cannot provision two or more users with one and the same email address – this will result in failure of the provisioning task.			
IS_PASSWORD_ENABLED	MX_PASSWORD_DISABLED	Boolean	This attribute is true when basic authentication is disabled.
IS_KERBEROS_ENABLED	MX_KERBEROS_ENABLED	Boolean	This attribute is true when Kerberos authentication is enabled.

HANA Attribute	SAP Identity Management Attribute	Mapping Format	Description
IS_SAML_ENABLED	MX_SAML_ENABLED	Boolean	This attribute is true when SAML authentication is enabled.
IS_X509_ENABLED	MX_X509_ENABLED	Boolean	This attribute is true when X509 authentication is enabled.
IS_SAP_LOGON_TICKET_ENABLED	MX_LOGON_TICKET_ENABLED	Boolean	This attribute is true when SAP logon ticket authentication is enabled.
IS_SAP_ASSERTION_TICKET_ENABLED	MX_ASSERTION_TICKET_ENABLED	Boolean	This attribute is true when authentication with SAP assertion ticket is enabled.
USER PARAMETERS	MX_USER_PARAMS	Multi-Value String	Except for e-mail address, locale, and timezone user parameters, the mapping for the general purpose user parameters follows the pattern <parameter name>==<value>. It replicates what the user has to enter in the SAP HANA system. ❖ Example CONTRACT_ID==89312642

HANA Attribute	SAP Identity Management Attribute	Mapping Format	Description
SAML USER MAPPING	MX_SAML_MAPPING	SAML mapping	The expected format for SAML mapping attribute is configured in the Identity Management User Interface. In the Change Identity task > Account attributes > SAML Mapping section you need to select <provider name> and its mapping <external identity>. If the desired mapping is to use the user from the target system, use the string <ANY> for an external identity. Example The following creates a mapping to username <i>John</i> in the Identity Management User Interface. It is displayed in a two-column table. The first column displays the provider name: AC_SAML_PROVIDER and the second column displays the external identity: John The following creates a mapping with the user from the SAP HANA Platform: Provider Name: CORP_SAML External Identity: <ANY>

HANA Attribute	SAP Identity Management Attribute	Mapping Format	Description
X509 USER MAPPING	MX_X509_MAPPING	Multi-Value String	The expected format for X509 mapping attribute is <subject DN>==<issuer DN>. It replicates what the user has to enter in the SAP HANA system. ❖ Example <pre> C=US, ST=California, L=San Francisco, O=Comapny A, OU=Unit A,CN=www.unita.org/ emailAddress=com panya@unita.org= =C=US,ST=Illinoi s, L=Chicago, O=Company B, OU=Certification Services Division, CN=Company B Server CA/ emailAddress=ser ver- certs@companyb.c om </pre>

On the user interface, these user attributes can be defined or modified for any users.

Related Information

[Setting Up an SAP HANA System \[page 632\]](#)

9.2.2.5 Mapping Between Identity Management and SuccessFactors Attributes

The following table shows the SuccessFactors attributes that are supported by the Success Factors connector and how they are mapped to attributes in the Identity Management.

User Attribute Mappings

SAP Identity Management Attribute	SuccessFactors OData Attribute	SuccessFactors SOAP Attribute
MSKEYVALUE	username	logon_user_name
MX_FIRSTNAME	firstName	personal_information/first_name
MX_LASTNAME	lastName	personal_information/last_name
MX_LANGUAGE	defaultLocale	
MX_MAIL_PRIMARY	email	email_information/email_address
MX_MOBILE_PRIMARY	cellPhone	phone_information/phone_number
MX_PHONE_PRIMARY	businessPhone	phone_information/phone_number
MX_HCM_SYSUNAME	username	logon_user_name
MX_ADDRESS_CITY	city	address_information/city To read city information, it must not be empty in SuccessFactors, and address_type must be set to business .
MX_ADDRESS_COUNTRY	country	address_information/country To read country information, it must not be empty in SuccessFactors, and address_type must be set to business .
MX_ADDRESS_STREETADDRESS	addressLine1	address_information/address1 To read address information, it must not be empty in SuccessFactors, and address_type must be set to business .
MX_FS_PERSONNEL_NUMBER	empld	
MX_FS_EMPLOYMENT_STATUS_ID		jobInformation/emplStatus
MX_FS_EMPLOYMENT_STATUS		Active or Inactive
MX_FS_COMPANY_CODE_ID	division	jobInformation/division
MX_FS_COMPANY_CODE	division	FO_division
MX_FS_ORGANIZATIONAL_UNIT_ID	department	jobInformation/department
MX_FS_ORGANIZATIONAL_UNIT	department	FO_department
MX_FS_POSITION_ID	title	jobInformation/job_title

SAP Identity Management Attribute	SuccessFactors OData Attribute	SuccessFactors SOAP Attribute
MX_FS_POSITION	title	jobInformation/job_title
MX_FS_JOB_ID	jobCode	jobInformation/job_code
MX_FS_JOB	jobCode	
MX_FS_PERSONNEL_AREA_ID	location	jobInformation/location
MX_FS_PERSONNEL_AREA	location	
MX_FS_CENTRALPERSON_ID	userId	login_user_id
MX_FS_PERSONNEL_NUMBER_OF_MANAGER	manager/userId	jobInformation/manager_id
MX_FS_IDENTITY_TYPE		EMPLOYEE
MX_FS_PERNR_IS_MANAGER	calculated	calculated
MX_DISABLED		employee status

Related Information

[Setting Up a SuccessFactors System \[page 640\]](#)

9.2.2.6 Mapping Between Identity Management and Domino Technical Attributes

The following tables show the Domino Technical attributes that are supported by the Lotus Notes connector and how they are mapped to attributes in the Identity Management.

User Attribute Mappings

Domino Technical Attribute	SAP Identity Management Attribute	Description
NotelD	MSKEYVALUE	Unique entry identifier
NotelD	ACCOUNT<Repository>	Unique user ID for the user in the target repository. The user has one attribute for each repository the user is added to.
FullName	MX_NOTES_FULLNAME	Entry's full name, which also represents entry's location. E.g. cn=Torkil Torkil-sen/o=sap
FullName	MX_NOTES_REGFULLNAME	User's full name entered during the initial user registration

Domino Technical Attribute	SAP Identity Management Attribute	Description
FullName	MX_NOTES_OLDFULLNAME	User's name before the name change
FirstName, MiddleName and LastName	DISPLAYNAME	Display name of an employee
FirstName	MX_FIRSTNAME	User's first name
MiddleInitial	MX_INITIALS	User's initials
LastName	MX_LASTNAME	User's last name
	MX_ENCRYPTED_PASSWORD	User's encrypted password
	MX_PASSWORD	User's password
ClientType	MX_NOTES_CLIENTTYPE	Maps the client type field in the Lotus Notes address book
MailAddress	MX_NOTES_MAILADDRESS	Holds the information from the mail address field in the Lotus Notes address book
ShortName	MX_NOTES_SHORTNAME	User's short name (e.g. TTork).
MailDomain	MX_NOTES_MAILDOMAIN	User's mail domain
MailFile	MX_NOTES_MAILFILE	User's mail file
FullName	MX_NOTES_ORGUNIT	User's organization unit
FullName	MX_NOTES_ORG	User's organization
FullName	MX_NOTES_COUNTRYCODE	Country code of the full name, where the user was created
NoteID	MX_NOTES_NOTEID	ID of the note (Lotus Notes object) on the Lotus Domino server
NoteUNID	MX_NOTES_UNID	Unique ID for the Lotus Notes object in Lotus Notes
InternetAddress	MX_MAIL_PRIMARY	User's primary e-mail address.
MailSystem	MX_NOTES_MAILSYSTEM	User's mail system, such as Lotus Notes, CcMail, Vim, etc.
Oracomment	DESCRIPTION	Entry description.
City	MX_ADDRESS_CITY	The city of the entry.
Country	MX_ADDRESS_COUNTRY	The country of the entry.

Domino Technical Attribute	SAP Identity Management Attribute	Description
Zip	MX_ADDRESS_POSTAL_CODE	The postal code of the entry.
StreetAddress	MX_ADDRESS_STREETADDRESS	The address of the entry.
PreferredLanguage	MX_LANGUAGE	The language of the entry.
Department	MX_DEPARTMENT	The department of the entry.
HomeFAXPhoneNumber	MX_FAX_PRIMARY	The primary fax number of the entry.
NetUserName	MX_LOGONALIAS	The alias for logon.
CellPhoneNumber	MX_MOBILE_PRIMARY	The entry's primary mobile number.
PhoneNumber_6	MX_PAGER_PRIMARY	The entry's primary pager number.
PhoneNumber	MX_PHONE_PRIMARY	The entry's primary telephone number.
Title	MX_SALUTATION	This attribute holds the title.
JobTitle	MX_TITLE	This attribute holds the title of the user.

Group Attribute Mappings

Domino Technical Attribute	SAP Identity Management Attribute	Description
ListName	MSKEYVALUE	Unique entry identifier.
ListName	DISPLAYNAME	Display name of a group.
NoteID	ACCOUNT<Repository>	Unique group ID for the group in the target repository. The group has one attribute for each repository the group is added to.
ListDescription	DESCRIPTION	Entry description.
GroupType	MX_NOTES_GROUP GROUPTYPE	Lotus Notes group types.

Note

The ListName attribute, which holds the name of the group in Lotus Notes, cannot be modified. In SAP Identity Management, it is used as the group's MSKEYVALUE, which is the unique entry identifier.

Domino Technical Attribute	SAP Identity Management Attribute	Description
GroupType	MX_NOTES_GROUP_GROUP-TYPE_DISPLAY	Lotus Notes group types in textual form (the textual form of the group type values held by the attribute MX_NOTES_GROUP_GROUPTYPE).
ListName	MX_NOTES_GROUP_LISTNAME	The name of the group in Lotus Notes.
NoteID	MX_NOTES_NOTEID	The ID of the note (Lotus Notes object) on the Lotus Domino server.
NoteUNID	MX_NOTES_UNID	The unique ID for the Lotus Notes object in Lotus Notes.

Related Information

[Setting Up a Lotus Notes System \[page 682\]](#)

9.2.2.7 Mapping Between Identity Management and Active Directory Attributes

The following tables show the Microsoft Active Directory attributes that are supported by the Active Directory connector and how they are mapped to attributes in the Identity Management.

User Attribute Mappings

Active Directory Attribute	SAP Identity Management Attribute	Description
userid	MSKEYVALUE	Unique entry identifier
mail	MX_MAIL_PRIMARY	User's primary e-mail address
givenName	MX_FIRSTNAME	User's first name
sn	MX_LASTNAME	User's last name
dn	ACCOUNT<Repository>	Unique user ID for the user in the target repository. For Active Directory, this should be the DN of the user. The user has one attribute for each repository the user is added to.
postofficebox	MX_ADDRESS_POBOX	User's PO box (Post Office box)

Active Directory Attribute	SAP Identity Management Attribute	Description
postalcode	MX_ADDRESS_POSTAL_CODE	User's postal code
street	MX_ADDRESS_STREETADDRESS	User's address
ou	MX_DEPARTMENT	User's department
description	DESCRIPTION	Entry description
displayname	DISPLAYNAME	Display name of an employee
facsimileTelephoneNumber	MX_FAX_PRIMARY	User's primary fax number
mobile	MX_MOBILE_PRIMARY	User's primary mobile number
telephoneNumber	MX_PHONE_PRIMARY	User's primary telephone number

Group Attribute Mappings

AD Attribute	SAP Identity Management Attribute	Description
dn	MSKEYVALUE	Unique entry identifier
cn	DISPLAYNAME	Display name of a group
	MX_REPOSITORYNAME	Repository name
dn	ACCOUNT<Repository>	Unique group ID for the group in the target repository. The group has one attribute for each repository the group is added to.

Related Information

[Setting Up a Microsoft Active Directory and Microsoft Exchange \[page 679\]](#)

9.2.2.8 Mapping Between Identity Management and Identity Authentication Service Attributes

The following table shows the SAP Cloud Identity Services - Identity Authentication attributes that are supported by the Identity Authentication connector and how they are mapped to attributes in the Identity Management.

User Attribute Mappings

SAP Identity Management Attributes	SAP Cloud Identity Attributes	Description
DISPLAYNAME	displayName	User friendly name
MSKEYVALUE	username or id	Unique entry (user) identifier
MX_ADDRESS_CITY	city	City
MX_ADDRESS_COUNTRY	country	Country key
MX_ADDRESS_POSTAL_CODE	postalCode	Postal code
MX_ADDRESS_REGION	region	Region
MX_ADDRESS_STREET_1	streetAddress	Street
MX_DEPARTMENT	department	Department
MX_DISABLED	active	User is disabled Boolean values User is not able to log on to Identity Management User Interface when disabled.
MX_ENCRYPTED_PASSWORD	password	Encrypted password used for password provisioning
MX_FIRSTNAME	firstName	User first name
MX_LASTNAME	lastName	User last name
MX_LANGUAGE	locale	User language
MX_MAIL_PRIMARY	email	Primary e-mail address
MX_PHONE_PRIMARY	businessPhone	Primary telephone number
MX_MOBILE_PRIMARY	cellPhone	Primary mobile number
MX_TITLE	title	Title of user

SAP Identity Management Attributes	SAP Cloud Identity Attributes	Description
MX_USER_UUID	userUuid	User UUID generated by Identity Authentication service
ACCOUNT<Repository>	id	Unique user ID for the user in the target repository. For SAP Cloud Identity service, this should be the id of the user. The user has one attribute for each repository the user is added to.

9.2.2.9 Mapping Between Identity Management and SCIM Attributes

The following table shows the SCIM 2.0 attributes that are supported by the SCIM connector and how they are mapped to attributes in the Identity Management. The SCIM connector supports the SCIM core schema attributes and enterprise user resource schema attributes.

User Attribute Mappings

SAP Identity Management Attributes	SCIM 2.0 Attributes	Description
DISPLAYNAME	displayName	User friendly name
MSKEYVALUE	userName	Unique entry (user) identifier
MX_ADDRESS_CITY	addresses.locality	City
MX_ADDRESS_COUNTRY	addresses.country	Country key
MX_ADDRESS_POSTAL_CODE	addresses.postalCode	Postal code
MX_ADDRESS_REGION	addresses.region	Region
MX_ADDRESS_STREET_1	addresses.streetAddress	Street
MX_DEPARTMENT	department	Department
MX_DISABLED	active	User is disabled Boolean values User is not able to log on to Identity Management User Interface when disabled.
MX_FIRSTNAME	name.givenName	User first name
MX_LASTNAME	name.familyName	User last name

SAP Identity Management Attributes	SCIM 2.0 Attributes	Description
MX_LANGUAGE	locale	User language
MX_MAIL_PRIMARY	emails.primary	Primary e-mail address
MX_PHONE_PRIMARY	phoneNumbers	Primary telephone number
MX_MOBILE_PRIMARY	phoneNumbers	Primary mobile number
MX_TITLE	title	The user's title, such as "Vice President".
ACCOUNT<Repository>	id	Unique user ID for the user in the target repository. The user has one attribute for each repository the user is added to.
MX_SF_ORGANIZATIONAL_UNIT	organization	Name of an organization

Group Attribute Mappings

SAP Identity Management Attributes	SCIM 2.0 Attributes	Description
MSKEYVALUE	id	
DISPLAYNAME	displayName	

9.2.2.10 Mapping Between Identity Management and SAP S/4HANA Attributes

The following table shows the SAP S/4HANA attributes that are supported by the S/4HANA connector and how they are mapped to attributes in SAP Identity Management.

Attributes that are explicitly used for the SAP S/4HANA system are indicated in the identity store schema using the prefix MX_S4HANA so that they can be recognized and provisioned accordingly.

Attribute Mappings

SAP S/4HANA Attribute	SAP Identity Management Attribute	Description
PersonExternalID or UserAssignmentUserID	MSKEYVALUE	Unique entry identifier

COMMON ATTRIBUTES

SAP S/4HANA Attribute	SAP Identity Management Attribute	Description
ValidityPeriodStartDate	MX_VALIDFROM	Start date of user's validity Format: YYYY-MM-DD
ValidityPeriodEndDate	MX_VALIDTO	End date of user's validity Format: YYYY-MM-DD
PersonUUID	MX_PERSONUUID	Person UUID
PersonExternalID	MX_S4HANA_PERSON_EXTERNAL_ID	Person External ID
PersonID	MX_S4HANA_PERSON_ID	Person ID
UserAssignmentUserID	MX_S4HANA_USERASSIGNMENT_USERID	User ID
BusinessPartnerRoleCode	MX_S4HANA_BUSINESS_PARTNER_ROLE_CODE	Business Partner Role Code Possible values: • BUP003 (Employee) • BBP010 (Freelancer) • BBP005 (Service Provider)
MarkedForArchivingIndicator	MX_S4HANA_MARKED_FOR_ARCHIVING	Mark for Archiving
PERSONAL INFO ATTRIBUTES		
PersonallInformationPersonFullName	DISPLAYNAME	Person full name
PersonallInformationFirstName	MX_FIRSTNAME	First name
PersonallInformationMiddleName	MX_MIDDLENAME	Middle name
PersonallInformationLastName	MX_LASTNAME	Last name
PersonallInformationBirthName	MX_BIRTHNAME	Birth name
PersonallInformationNickName	MX_NICKNAME	Nick name
PersonallInformationInitials	MX_INITIALS	Initials
PersonallInformationAcademicTitle	MX_ACADEMIC_TITLE_1	Academic title
PersonallInformationAcademicSecondTitle	MX_ACADEMIC_TITLE_2	Academic second title

SAP S/4HANA Attribute	SAP Identity Management Attribute	Description
PersonallInformationCorrespondence-Language	MX_LANGUAGE	Correspondence language
PersonallInformationFormOfAddress	MX_S4HANA_FORM_OF_ADDRESS	Form of address
PersonallInformationAdditionalLast-Name	MX_S4HANA_ADDITIONAL_LAST-NAME	Additional last name
PersonallInformationLastNamePrefix	MX_S4HANA_LASTNAME_PREFIXS	Last name prefix
PersonallInformationLastNameSecond-Prefix	MX_S4HANA_LASTNAME_SEC-OND_PREFIXS	Last name second prefix
PersonallInformationNameSupplement	MX_S4HANA_NAME_SUPPLEMENT	Name supplement, for example noble title
WORKPLACE ATTRIBUTES		
WorkplaceInformationEmailAddress	MX_MAIL_PRIMARY	Email address
WorkplaceInformationDepartment	MX_DEPARTMENT	Department name
WorkplaceInformationRoomNumber	MX_WORKPLACE_ROOM	Room number
WorkplaceInformationBuilding	MX_WORKPLACE_BUILDING	Building name
WorkplaceInformationFunctionalTitle-Name	MX_S4HANA_FUNCTIONAL_TIT-LE_NAME	Functional title name
PHONE ATTRIBUTES		
primaryBusinessPhone	MX_PHONE_PRIMARY	In S/4HANA, a business partner can have only one business and one cell phone number. MX_PHONE_PRIMARY holds the business phone number. MX_MOBILE_PRIMARY holds the cell phone. The phone numbers in S/4HANA are transformed into phone strings in SAP Identity Management in the following format: <Country Code>:<Country Dialing Code> (<AreaID><SubscriberID>-<Extension> For example: BG:+359 (02)0700-1717

SAP S/4HANA Attribute	SAP Identity Management Attribute	Description
primaryCellPhone	MX_MOBILE_PRIMARY	DE : +49 (62277)16717 BG : +359 897123456 The phone fields <i>CountryDialingCode</i>, <i>AreaID</i>, <i>SubscriberID</i> and <i>Extension</i> are coming from the S/4HANA phone number JAVA objects. Note The S/4HANA phone numbers of type cell contain only the <i>CountryDialingCode</i> and the <i>SubscriberID</i>.

9.2.3 Attributes that Support Value Help

Attributes

Identity Center Attribute	Value Help Content	Modifiable / To be Read from AS ABAP	Language Dependency
MX_DATEFORMAT	Fixed (Delivered with the provisioning framework)	N	Y
MX_LANGUAGE	Fixed (Delivered with the provisioning framework)	N	Y
MX_LANGUAGE_COUNTRY	Fixed (Delivered with the provisioning framework)	N	Y
MX_NUMBERFORMAT	Fixed (Delivered with the provisioning framework)	N	Y
MX_TIMEZONE	Fixed (Delivered with the provisioning framework)	N	Y
MX_USERTYPE	Fixed (Delivered with the provisioning framework)	N	Y
MX_ACADEMIC_TITLE_1	Tables TSAD2 and TSAD2T	Y	N
MX_ACADEMIC_TITLE_2	Tables TSAD2 and TSAD2T	Y	N
MX_ADMIN_UNIT	Tables USGRP and USGRPT	Y	N

Identity Center Attribute	Value Help Content	Modifiable / To be Read from AS ABAP	Language Dependency
MX_NAME_PREFIX_1	Table TSAD4	Y	N
MX_NAME_PREFIX_2	Table TSAD4	Y	N
MX_PRINTERSET- TINGS_SPLD	Table TSP03	Y	N
MX_SALUTATION	Tables TSAD3 and TSAD3T	Y	N
MX_TITLE_SUPPLEMENT	Tables TSAD5 and TSAD5T	Y	N
MX_USER_CATEGORY	Tables USGRP and USGRPT	Y	N

The value help content is stored in the `mx_i_AttrValueHelp` database table, which is specified in the attribute value properties. See the example below for the `MX_DATEFORMAT` attribute.

General | Storage | Presentation | Validation | Event Processes | **Attribute Values** | Entry Types

Legal Attribute Values

☐ Value: Add

☐ SQL query: Modify...
Delete

☒ Value Help
 Table Name:
 Values ID:
☒ Language Dependent

☐ Task: ...

See also *Reading Value Help Content*.

Related Information

[Reading Value Help Content \[page 598\]](#)

9.2.4 Configuring the ABAP Connector to Use SNC

Prerequisites

- You have access to the SAP Cryptographic Library.

Note

The distribution of the SAP Cryptographic Library is subject to and controlled by German export regulations and is not available to all customers. In addition, the library may be subject to local regulations of your own country that may further restrict the import, use and (re-)export of cryptographic software. If you have any further questions on this issue, contact your local SAP subsidiary.

- The AS ABAP is configured to use SNC.
For more information, see the SNC documentation on the Help Portal.

Note

In the procedure below you exchange the public-key certificates.

Context

You can use Secure Network Communications (SNC) to secure the connection between the Identity Center and the ABAP system.

SNC requires the use of an external security product to perform the security functions. For this purpose, you can use the SAP Cryptographic Library or any SNC-certified product. The following description shows how to configure SNC when using the SAP Cryptographic Library. If you are using a different SNC-certified product, then see the documentation provided by the vendor for information about how to establish the security context for the Identity Center.

Procedure

1. Download and install the SAP Cryptographic Library.
2. Create a Personal Security Environment for the Identity Center.
3. Create credentials for the Identity Center.
4. Exchange the public-key certificates belonging to the Identity Center and the AS ABAP.
5. Set the SNC parameters for the AS ABAP connector.
6. Maintain the extended user access control lists (ACL) on the AS ABAP to allow the service user to connect to the AS ABAP using the SNC connection.

7. Test the connection.

Related Information

[Secure Network Communications \(SNC\)](#)
[Creating a Personal Security Environment \[page 1314\]](#)
[Creating Credentials \[page 1316\]](#)
[Exchanging the Public-Key Certificates \[page 1318\]](#)
[Setting the SNC Parameters \[page 1323\]](#)
[Maintaining the Extended User ACL \[page 1325\]](#)
[Testing the Connection \[page 1325\]](#)
[Downloading and Installing the SAP Cryptographic Library \[page 1328\]](#)

9.2.4.1 Creating a Personal Security Environment

Context

Use the command line tool's command `get_pse` to generate the server's PSE, which includes the public and private key pair and a public-key certificate. If you are using a trusted CA, then you can also use the `get_pse` command to generate a certificate request. Per default, all of the items are generated, however, you can use the options `-noreq` or `-onlyreq` to explicitly include or omit the certificate request.

→ Tip

For easier administration, we recommend using self-signed certificates that are not signed by a trusted CA.

📘 Note

As an alternative to creating a PSE, you can use the same PSE as the AS ABAP server. If this is the case, and you have already created the PSE on the AS ABAP, then copy the AS ABAP's PSE to the appropriate location instead of creating a new one.

Procedure

1. Start a command line interface.
2. Navigate to the location of the `sapgenpse` command line tool.
3. Use the following command line to generate a PSE. Create the server's PSE in the SECUDIR directory.

sapgenpse get_pse [-p <PSE_name>] [-x <PIN>] [DN]

Where:

Command Line Options

Option	Parameter	Description	Allowed Values	Default
-p	<PSE_name>	Path and file name for the server's PSE	Path description (in quotation marks, if spaces exist)	None
-x	<PIN>	PIN that protects the PSE	Character string	None
None	DN	Distinguished Name for the server The Distinguished Name is used to build the server's SNC name.	Character string (in quotation marks, if spaces exist)	None

Note

The Distinguished Name consists of the following elements:

- CN = <Common_Name>
- OU = <Organizational_Unit>
- O = <Organization>
- C = <Country>

The following examples show possible Distinguished Names for the Identity Center:

- CN=IC, OU=MyDept, O=MyCompany, C=DE
- CN=IC, OU=IdM, O=MyCompany, C=US

There are also additional command line options that you can use to specify further details. See the table below.

Additional Command Line Options

Option	Parameter	Description	Allowed Values	Default
-r	<file_name>	File name for a certificate request	Path description (in quotation marks, if spaces exist)	stdout
-s	<key_len>	Key length	512, 1024, 2048	1024
-a	<algorithm>	Algorithm used	RSA, DSA	RSA

Option	Parameter	Description	Allowed Values	Default
-noreq	None	Only generate a key pair and PSE. Do not generate a certificate request.	Not applicable	Not set
-onlyreq	None	Generate a certificate request for the public key stored in the PSE specified by the -p parameter.	Not applicable	Not set

Results

The server's PSE is created in the directory you specified.

Note

Check the contents of the directory at the operating system level to make sure the PSE was created in the correct location before proceeding with the next step.

Example

The following command line generates a PSE for the Identity Center where a self-signed certificate is used.

```
sapgenpse get_pse -p IC.pse -noreq -x abcpin "CN=IC, O=MyCompany, C=DE"
```

9.2.4.2 Creating Credentials

Context

The server must have active credentials at runtime. Therefore, to produce active credentials, use the configuration tool's command `sec login` to "open" the server's PSE.

Note

The credentials are located in the `cred_v2` file in the directory specified in the `<SECUDIR>` environment variable. Make sure that only the user under which the server runs has access to this file (including read access).

⚠ Caution

It is very important to create the credentials for the user who runs the Identity Center's processes. In a default installation, this user is SYSTEM.

Procedure

Use the following command line to open the server's PSE and create credentials:

```
sapgenpse seclogin [-p <PSE_name>] [-x <PIN>] [-o [<NT_Domain>\]<user_ID>]
```

Where:

Command Line Options

Option	Parameter	Description	Allowed Values	Default
-p	<PSE_name>	Path and file name for the server's PSE	Path description (in quotation marks, if spaces exist)	None
-x	<PIN>	PIN that protects the PSE	Character string	None
-o	[<NT_Domain>] \<user_ID>	User for which the credentials are created. (The user that runs the dispatcher service.)	Valid operating system user	The current user

With the additional options, you can also use the seclogin command to delete the server's credentials, change the PIN that protects a PSE, or to list the available credentials for a user. See the table below.

Additional Command Line Options

Option	Parameter	Description	Allowed Values	Default
-l	None	List all available credentials for the current user.	Not applicable	Not set
-d	None	Delete PSE	Not applicable	Not set
-chpin	None	Specifies that you want to change the PIN	Not applicable	Not set

Results

The credentials file (cred_v2) for the user provided with the -o option is created in the SECUDIR directory.

Note

Check the contents of the directory at the operating system level to make sure the credentials were created in the correct location before proceeding with the next step.

Example

The following command line opens the Identity Center's PSE and creates credentials for the MXDispatcher_Service_User user.

```
sapgenpse seclogin -p IC.pse -x abcpin -0 MXDispatcher_Service_User
```

9.2.4.3 Exchanging the Public-Key Certificates

Context

This procedure assumes that you are using different PSEs for the AS ABAP server and the Identity Center.

If you are using the same PSE, then you can skip this step. In this step, you establish a trust relationship between the two servers by exchanging their public-key certificates. This procedure consists of the following steps.

Procedure

1. Export the Identity Center's public-key certificate from the Identity Center's PSE.
2. Import the Identity Center's public-key certificate into the AS ABAP's SNC PSE.
3. Export the AS ABAP's public-key certificate from the AS ABAP's SNC PSE.
4. Import the AS ABAP's public-key certificate into the Identity Center's PSE.

9.2.4.3.1 Exporting the Identity Center's Public-Key Certificate

Context

Procedure

Use the tool's command `export_own_cert` to export the server's certificate:

```
sapgenpse export_own_cert -o <output_file> -p <PSE_name> [- x <PIN>]
```

Where:

Command Line Options

Option	Parameter	Description	Allowed Values	Default
-o	<output_file>	Exports the certificate to the named file	Path description (in quotation marks, if spaces exist)	stdout
-p	<PSE_name>	Path and file name for the server's PSE	Path description (in quotation marks, if spaces exist)	None
-x	<PIN>	PIN that protects the PSE	Character string	None

Example

The following command line exports the Identity Center's public-key certificate to the `IC.crt` file.

```
sapgenpse export_own_cert -o IC.crt -p IC.pse -x abcpin
```

9.2.4.3.2 Importing the Identity Center's Public-Key Certificate Into the AS ABAP's SNC PSE

Prerequisites

- You know the PIN that protects access to the AS ABAP's SNC PSE.
- You have access to the Identity Center's public-key certificate that you exported in the last step.

Context

If the AS ABAP uses the SAP Cryptographic Library as its security provider for SNC, then you can use the trust manager to maintain the AS ABAP's SNC PSE.

Note

Using trust manager (transaction STRUST), as described in the procedure below, is just one of the possible options to exchange public-key certificates between the Identity Center and AS ABAP.

Using the trust manager on the AS ABAP (transaction STRUST), proceed as follows:

Procedure

1. Select the *SNC SAPCryptolib* PSE with a double-click.
2. Enter the PIN that protects access to the PSE.

Information about the SNC PSE appears in the upper section of the trust manager's screen.

3. Choose  *Certificate*  *Import*  from the menu or the symbol for *Import certificate*.
4. In the dialog that follows, enter the path and file name of the Identity Center's public-key certificate file, select the *Base64* format, and choose *Enter*.

The certificate appears in the *Certificate* section of the trust manager's screen.

5. Choose *Add to Certificate List* to add the certificate to the AS ABAP's SNC PSE.
6. Save the data.

Caution

Do not forget to save the data. Otherwise, changes made to the PSE are lost.

9.2.4.3.3 Exporting the AS ABAP's Public-Key Certificate

Context

Continue with exporting the AS ABAP's public-key certificate.

Note

Using trust manager (transaction **STRUST**), as described in the procedure below, is just one of the possible options to exchange public-key certificates between the Identity Center and AS ABAP.

Procedure

1. Make sure the *SNC SAPCryptolib* is still the selected PSE.
2. Select the certificate shown in the *Owner* field with a double-click.
Information about the certificate appears in the *Certificate* section.
3. Choose  *Certificate*  *Export*  from the menu or the symbol for *Export certificate*.
4. In the dialog that follows, enter the path and file name where you want to save the file, select the *Base64* format and choose *Enter*.

The file is saved to the file system.

Results

You can now import this file into the Identity Center's PSEs.

9.2.4.3.4 Importing the AS ABAP's Public-Key Certificate Into the Identity Center's PSE

Return to the Identity Center server and import the AS ABAP's public-key certificate into the Identity Center's PSE.

Prerequisites

- You have access to the AS ABAP's public-key certificate that you exported in the last step.

Context

Procedure

Use the tool's command `maintain_pk` to import the AS ABAP's public-key certificate into the Identity Center PSE's certificate list.

```
sapgenpse maintain_pk [-a <cert_file>] -p <PSE_name> [-x <PIN>]
```

Where:

Command Line Options

Option	Parameter	Description	Allowed Values	Default
-a	<cert_file>	Add certificate from file <cert_file> to the certificate list.	Path description (in quotation marks, if spaces exist)	None
-p	<PSE_name>	Path and file name for the server's PSE	Path description (in quotation marks, if spaces exist)	None
-x	<PIN>	PIN that protects the PSE	Character string	None

Additional Command Line Options

Option	Parameter	Description	Allowed Values	Default
-m	<cert_file>	Add multiple certificates from <cert_file> to the certificate list.	Not applicable	None
-M	<store>	Add multiple certificates from the CryptoAPI certificate store to the certificate list.	ROOT, CA, MY, SPC	None
-d	<number>	Delete certificate number <number> from certificate list.	Numerical value	None
-l	None	List existing certificate list.	Not applicable	None
-y	None	Automatic YES-mode for -m or -M options.	Not applicable	None

Example

The following command line imports the AS ABAP's certificate from the <AS_ABAP_DIR_INSTANCE>\sec\ABC.crt file into the Identity Center's PSE.

```
sapgenpse maintain_pk -a <AS_ABAP_DIR_INSTANCE>\sec\ABC.crt -p  
<Install_folder>\sec\IC.pse
```

9.2.4.4 Setting the SNC Parameters

Context

You then have to set the connection-specific SNC parameters in the repository constants for the AS ABAP system that you are connecting to.

Procedure

Set the parameters as shown in the table below.

Repository Constants for SNC

Parameter	Description	Permitted Values	Example
JCO_CLIENT_ SNC_LIB	Path and file name of the SAP Cryptographic Library	String value	<Install_folder>\sap-crypto.dll
JCO_CLIENT_ SNC_MODE	SNC activation indicator	0, 1 0 = SNC disabled 1= SNC activated	1
JCO_CLIENT_ SNC_MYNAME	The Identity Center's SNC name.	String value	p:CN=IC, O=MyCompany, C=DE
JCO_CLIENT_ SNC_PARTNERNAME	SNC name of the communication partner (AS ABAP)	String value	p:CN=ABC, O=MyCompany, C=DE
JCO_CLIENT_ SNC_QOP	Quality of protection level	1: Secure authentication only 2: Data integrity protection 3: Data privacy protection 9: Use the value from <code>snc/data_protection/max</code> .	3

For more information about setting repository constants, see *Creating Repositories*. For more information about transporting repository definitions, see the *Transport Guide* on the SAP Community Network. For more information about the SNC parameters, see the SAP Help Portal.

Related Information

[SAP Identity Management Implementation Guide - Transport](#) 

[SAP Help Portal](#)

[Creating Repositories \[page 493\]](#)

9.2.4.5 Maintaining the Extended User ACL

Context

When setting up the SNC-protected connection, the two systems are authenticated within the SNC layer, but not the actual user that is to log on to the AS ABAP using the connection. Therefore, to allow the service user to connect to the AS ABAP using the secure connection, you must maintain the extended user ACL. Proceed as follows:

Procedure

1. Using table maintenance (for example, transaction SM30), maintain the USRACLEXT table.
2. Choose [New Entries](#).
3. Enter the following data in the corresponding fields:

Required Data

Field	Value	Example
User	<IC_service_user>	IC_SERV_USER
Sequence Number	<sequential number>	000
SNC Name	<IC_SNC_Name>	p:CN=IC, O=MyCompany, C=DE

4. Save the data.

9.2.4.6 Testing the Connection

Context

Procedure

1. Set up a job that reads data from the AS ABAP and run it.
2. If the job returns errors, set the following system environment variables:
 - RFC_TRACE = 1
 - CPIC_TRACE = 3
3. Run the job again.
4. Check the CPIC trace file.

This file has the name `CPICTRC<number>` and, by default, is located in the Identity Center's install folder.

The file contains the SNC initialization information. A correct initialization appears as follows.

```
[Thr 3560] <<- SncSetParam()==SAP_O_K
[Thr 3560] ->> SncInit(prg=5, ini_fname=(NULL),
&sec_avail=000784D3)
[Thr 3560] SncInit(): Initializing Secure Network Communication
(SNC)
[Thr 3560] PC with Windows NT
(mt,ascii,SAP_UC/size_t/void* = 8/32/32)
[Thr 3560] SncInit(): Trying user/application supplied as a
gssapi library name: "C:\Program Files\SAP\IdM\Identity
Center\sapcrypto.dll".
[Thr 3560] load shared library (C:\Program
Files\SAP\IdM\Identity
Center\sapcrypto.dll), hdl 0
[Thr 3560] using "C:\Program Files\SAP\IdM\Identity
Center\sapcrypto.dll"
...
[Thr 3560] File "C:\Program Files\SAP\IdM\Identity
Center\sapcrypto.dll" dynamically loaded as GSS-API v2
library.
[Thr 3560] The internal Adapter for the loaded GSS-API
mechanism identifies as:
Internal SNC-Adapter (Rev 1.0) to SECUDE 5/GSS-API v2
[Thr 3560] <<- SncPDLInit()==SAP_O_K
[Thr 3560] SncInit(): Initiating Credentials available,
lifetime=263246h 32m 39s
[Thr 3560] <<- SncInit()==SAP_O_K
[Thr 3560] sec_avail = "true"
[Thr 3560] ->> SncSessionInit(&snc_hdl=000784D4)
[Thr 3560] <<- SncSessionInit()==SAP_O_K
[Thr 3560] out: &snc_hdl = 0D753FC8
[Thr 3560] ->> SncSetMyName(snc_hdl=0D753FC8, myname="p:CN=IC,
O=MyCompany, C=DE")
[Thr 3560] <<- SncSetMyName()==SAP_O_K
[Thr 3560] in: myname = "p:CN=IC, O=MyCompany, C=DE"
[Thr 3560] ->> SncSessionInitiator(snc_hdl=0D753FC8,
auth_type=1, buf_size_hint=0,target='p:CN=ABC,
O=MyCompany, C=DE')
[Thr 3560] <<- SncSessionInitiator()==SAP_O_K
[Thr 3560] in: target = "p:CN=ABC, O=MyCompany, C=DE"
[Thr 3560] parses to = "p:CN=ABC, O=MyCompany, C=DE"
[Thr 3560] ->> SncSetQOP(snc_hdl=0D753FC8, min=max default,
max=max default, qop=max default)
[Thr 3560] <<- SncSetQOP()==SAP_O_K
[Thr 3560] in: qop values = "min=9 (max default), max=9
(max default), use=9 (max default)"
[Thr 3560] resulting = "min=3 (old:2), max=3 (old:3),
use=3 (old:3)"
[Thr 3560] STISncInit: set snc state to SNC_ENABLED
```

Any errors will be indicated in this initialization block. The table below shows typical error conditions.

Typical Errors

Error	Description	Solution
GSS-API(maj): No credentials were supplied GSS-API(min): Can't read file Couldn't acquire DEFAULT INITIATING credentials	Either credentials were not created, or they were created for the wrong user.	Make sure the Identity Center's SNC is correct. Use the sapgenpse command <code>get_my_name</code> to obtain the Distinguished Name being used. Make sure this coincides with the repository constant <code>JCO_CLIENT_SNC_MY-NAME</code>. Also make sure the credentials exist for the correct user. To make sure, delete the <code>cred_v2</code> file (if it exists) and create credentials for the correct user. (The user ID is also indicated in the error message.) Use the sapgenpse <code>se-clogin</code> command with the <code>-O</code> option to set the user for which the credentials apply.
SncSetMyName()== SNCERR_BAD_NT_PREFIX in: my-name in: target	The message in: myname indicates that the Identity Center's SNC name is not correct. The message in: target indicates that the AS ABAP's SNC name is not correct.	
GSS-API(maj): A token had an invalid signature GSS-API(min):The name is wrong	The connection could not be established.	Check the value of the repository constant <code>JCO_CLIENT_SNC_MY-NAME</code> or <code>JCO_CLIENT_SNC_PARTNERNAME</code> depending on the corresponding message. Make sure the SNC name contains the prefix <code>p:</code> .
GSS-API(maj): A token had an invalid signature GSS-API(min): Certification path ends at wrong CA	The connection could not be established.	Check the SNC name for the AS ABAP in the repository constant <code>JCO_CLIENT_SNC_PARTNERNAME</code> . Make sure it corresponds to the SNC name specified in the profile parameter <code>snc/identity/as</code> on the AS ABAP.

Error	Description	Solution
no conversation found with id <ID> In addition, the error SNC name and specified user/client do not match appears in the job log.	The connection could not be established with the given parameter values.	The certification path could not be verified. Check the trust relationships. Make sure the Identity Center's publickey certificate is contained in the SNC PSE's certificate list on the AS ABAP. Also make sure the AS ABAP's publickey certificate is contained in the Identity Center's certificate list. You can use the sapgenpse maintain_pk – l option to check the Identity Center's certificate list.
SNC disabled, reject request from host=<host> TP=java	SNC is not active on target system.	Activate SNC and check the configuration on the target system.

9.2.4.7 Downloading and Installing the SAP Cryptographic Library

Prerequisites

- You know the user under which the dispatcher runs. You can find this user by checking the user that runs the Windows service MXDispatcher_<dispatcher_name>.

Context

The SAP Cryptographic Library installation package is available for authorized customers on the SAP Service Marketplace at service.sap.com/swdc.

The installation package sapcrypto.car contains the following files:

- The SAP Cryptographic Library (sapcrypto.dll for Windows NT or libsapcrypto.<ext> for UNIX).
- Depending on the version of SAP Cryptographic Library, there might be a corresponding license ticket (ticket).
- The configuration tool sapgenpse.exe.

Procedure

1. Download the file from the SAP Service Marketplace and extract it to a local directory.

If you are not authorized to download the file, then contact your local subsidiary to clarify whether you are allowed to receive the installation package.

2. Copy the library (`sapcrypto.dll`) and the command line tool (`sapgenpse.exe`) to a local directory, for example, the Identity Center's install directory.

`<Install_folder>\sapcrypto.dll`

`<Install_folder>\sapgenpse.exe`

3. (Optional) Copy the license ticket (ticket) to a local directory.

→ Tip

We recommend creating a subdirectory `sec` and copying the ticket to this directory.

`<Install_folder>\sec\ticket`

This will also be the location for the Identity Center's Personal Security Environment (PSE) that contains the key pair used for securing the connections.

4. Set the SECUDIR environment variable to this directory.

`SECUDIR=<Install_folder>\sec`

5. Make sure you set SECUDIR for the user that runs the corresponding dispatcher (or as a system variable, if the user is the SYSTEM user).

ⓘ Note

Default SECUDIR

If SECUDIR is not set, then the server searches for the license ticket in the `sec` subdirectory of the server's home directory:

On Windows NT: `$HOMEDRIVE$HOMEPATH\sec\`

6. If the user running the dispatcher is different from the logged on user, set the system environment variable `USER` to the dispatcher's user.

10 Restrictions

10.1 Restrictions and Considerations that Apply to Specific Use Cases

10.1.1 Restrictions When Using the SAP HCM Use Case

The following restrictions apply when using the SAP HCM use case:

- The delta mechanism is not preconfigured when importing the data from the SAP HCM system into the staging area in the Identity Center. A full load is always performed.
- When using the SAP HCM use case, the following options are delivered with the SAP provisioning framework for determining the user account name:
 - The SAP HCM system can determine the user account name using the [P0105-SYHR_A_P0105_AF_SYSUNAME](#) field. If this field does not contain any data, but other employee data is maintained, then the employee data is first only written to the staging area. Only after this field is filled in the SAP HCM system is the complete data written to the productive identity store. Data consistency is ensured (for example, if you delete the user name from this field) by making sure that a unique personnel number is also specified in the [P0000-PERNR](#) field.
 - The Identity Center can determine the user account name. In this case, it uses the SAP HCM [P0000-PERNR](#) field as input to determine a unique system-wide user ID. The [P0105-SYHR_A_P0105_AF_SYSUNAME](#) field is ignored.

If you have other needs for determining the system-wide user account name, then you must adjust the tasks and jobs accordingly.

- For changes to attributes that should occur in the future, a pending object is created and then processed at the time of the change. By default, multiple changes to the same attribute produce multiple pending change request, with an exception for the `MX_FS_EMPLOYMENT_STATUS_ID`. For this attribute, an existing pending value is overwritten. If you want to change the behavior for this or other attributes, you must change the script that is used in the SAP HCM staging area identity store framework. For more information, see SAP Note [1524813](#).
- If you have difficulties transferring Unicode characters from the SAP HCM system, then start the system's LDAP connector using the code page that corresponds to the SAP HCM system. For more information, see SAP Note [539198](#).

10.1.2 Restrictions When Using the SAP Enterprise Portal

SAP Identity Management does not support remote roles used by a Federated Portal Network (FPN) to share content. If you are integrating a FPN with SAP Identity Management, use the Remote Delta Link mode to share content instead. For more information, see the Related Links section.

Related Information

['Remote Delta Link' Mode](#)

[FPN Part III - Sharing Content between SAP NetWeaver Portals](#) 

10.1.3 Restrictions When Using the Identity Lifecycle Management Use Case

When using this use case you must ensure that users exist and are assigned to the appropriate groups on the LDAP directory server before running jobs or initiating provisioning steps that will assign portal roles to the users. Otherwise, if a user exists in SAP HCM and is assigned to a portal role, and the portal role assignment is provisioned without the user existing in the LDAP directory server that is used as a user store for the portal, then you will receive errors.

10.1.4 Restrictions When Using the Enhanced SAP Business Suite Integration Use Case

For this case, note the following:

- There is no support for central user administration with this use case.
- When integrating SAP Identity Management with the SAP Business Suite, application-specific identity data is also provisioned. To ensure that the correct data is provisioned based on the systems that are integrated, there are application-specific Business Add-Ins (BADIs) to use for preprocessing and postprocessing. These BADIs are handled as privileges in the Identity Center as type FUNCTION_SET. You need to include them in the business roles that apply to the corresponding SAP Business Suite applications.
- By default, the Identity Center also maintains user account data when provisioning. For certain applications, you may want to have certain identity data provisioned without having a user account created for the identity, for example, when provisioning business partner data. For this case, create a dedicated repository for the AS ABAP where the user account should not be created. For this repository, specify the NO_USER_ACCOUNT repository constant .

10.2 Restrictions and Considerations that Apply to Specific Connectors

10.2.1 Restrictions that Apply to All Connectors

The following restrictions apply when using the SAP provisioning framework:

- The users used for the connections should be technical users that do not have to change their passwords, for example, service users in AS ABAP or technical users in AS Java.

⚠ Caution

Since the connections are system-to-system connections that do not have a user interface, if the user is a dialog user and is required to change his or her password, for example, if the password is initial, then errors will occur.

ℹ Note

The administrator of the LDAP directory must create a user that SAP Identity Management can use to connect to the LDAP server. This user should have read and search permissions as well as create and change authorizations for all branches of the LDAP directory.

- Whenever a user attribute is changed (except for role and privilege assignments), all user attributes are provisioned to the selected back-end systems (not only the changed attributes). If you want to remove attributes from the set of attributes to provision, modify the *Update System Privilege trigger attributes* pass in the initial load job for the appropriate connected system. This pass contains a negative list of attributes that are not to be provisioned to the target systems.
- To deactivate (or completely delete) a user in the SAP Identity Management identity store and in all target systems, first remove all privileges from the user assignments. This will delete the user in the individual repositories. After that, you can deactivate or delete the user in the identity store. If you deactivate or delete a user without removing the privilege assignments, the user will be deleted in the identity store, but not in the individual repositories.
By default, the web-enabled task *Inactivate Identity* is configured to only deactivate a user in the identity store and not delete it. In this way, the history of the user can be retained. If you want to actually delete the identity, adjust the pass accordingly. (Set MX_INACTIVE to FALSE in the *Destination* tab for the *Inactivate Identity* pass.)
- When performing the initial loads, consolidation occurs based on user IDs, meaning that an identity is created in the identity store for each unique user ID that is read.
- Company addresses are read from the system that is specified as the leading system for company addresses and provisioned to the rest of the connected systems. This means you cannot maintain system-dependent company addresses throughout your landscape.

Related Information

[Setting Up an AS ABAP System \[page 623\]](#)

10.2.2 Restrictions for AS ABAP/SAP Business Suite System Connectors

- Time dependencies for privilege assignments are read into the Identity Center with the initial load. The privilege assignments are then provisioned to the target systems when they become active.

⚠ Caution

After the initial load, the time dependencies are stored in the Identity Center and no longer in the AS ABAP. Previous time-dependent assignments are lost in this step, therefore, you no longer have a history of such assignments. You also no longer see future assignments in the AS ABAP.

To improve efficiency, you can execute the PRGN_COMPRESS_TIMES report with the [Remove Validity Periods That Have Already Expired](#) option for all users. This removes all outdated role assignments so that the initial load only reads active and future role assignments.

- The ABAP connector does not support reference users.
- In AS ABAP systems, a new Customizing switch for local user management defines the user group as mandatory field for specific clients (see SAP Note [1663177](#)). Therefore, users can no longer be created without entering a valid user group. If you activate this switch for any AS ABAP repository, you must ensure that for each user a corresponding user group is available, for example, using an entry field in your customized user interface.
- Composite roles and derived roles are read into the identity store, however, there is no information in the Identity Center to indicate these role types. In the Identity Center, you see a flat list containing all roles.
- When you use SAP Identity Management, you should not assign roles and profiles locally in your AS ABAP systems any longer. All local assignments of roles to users are overwritten by any change in the user's central assignments carried out with SAP Identity Management. If you change any assignment there, a list of all assigned roles and privileges for the user is sent to the relevant AS ABAP system overwriting all local data in the AS ABAP system. To synchronize the authorization assignments in SAP Identity Management and the local AS ABAP, we recommend to use reconciliation jobs.
- Whenever a role or group assignment is changed, all role, profile, and group assignments are provisioned (not just the delta). The assignments are provisioned to all systems that are affected by the change.
- The Provisioning framework for SAP Identity Management does not support indirect role assignments coming from SAP Human Capital Management (HCM). If you want to manage the roles on a per user bases in SAP Identity Management, you need to assign the roles directly to a user. Besides, all the role assignments provisioned from SAP Identity Management will be direct assignments.
- Mobile numbers must not contain a hyphen (-). The ABAP connector interprets the hyphen (-) as an extension, but the AS ABAP ignores extensions for mobile numbers.
- When connecting to an AS ABAP Release 4.6C, the password used for the connection needs to be entered in the repository constants in upper case.
- Not all identity attributes are supported, for example, licensing attributes. See *Appendix B: Mapping Between Identity Center and AS ABAP Attributes* for a list of the supported attributes.

Related Information

[Mapping Between Identity Management and AS ABAP Attributes \[page 1277\]](#)

[Setting Up an AS ABAP System \[page 623\]](#)

10.2.3 Restrictions when Connecting a CUA System

- To support a CUA landscape, connect the CUA central system to the Identity Center using the ABAP connector. The Identity Center provisions identity data to the CUA central system, which in turn provisions the data to its child systems. This provisioning takes place according to the configuration of the attribute distribution settings on the central system.
- The CUA is not supported with the SAP Business Suite integration use case.
- Using both integration scenarios - SAP Identity Management with CUA and SAP Identity Management with SAP Governance, Risk and Compliance (GRC) is not supported. You can use only one of them in your system landscape.
- Although you do not have to change the attribute distribution settings (using transaction SCUM), we recommend using the global distribution setting for attributes so that they can be maintained in the Identity Center.
- Only connect the CUA central system to the Identity Center. Do not connect any of the CUA child systems. If you want to connect a child system directly to the Identity Center, disconnect it from CUA first.
- If a corresponding LDAP directory is also connected to the Identity Center, then the LDAP synchronization for the CUA central system is obsolete.
- You no longer need to assign users to systems in the CUA landscape as the Identity Center makes this correlation when a user is assigned a privilege in the corresponding system.
- In case a CUA managed system is connected, note that the system will try to deprovision privileges when their validity is reached. Because of this, the deprovisioning process will continue being triggered until the privileges are removed successfully.

Related Information

[Integrating a Central User Administration System \[page 32\]](#)

10.2.4 Restrictions for AS Java System Connectors

- Whenever a role or group assignment is changed, all role, profile, and group assignments are provisioned (not just the delta). The assignments are provisioned to all systems that are affected by the change.
- For AS Java with an LDAP directory as the UME user store, note that the UME group assignments are not provisioned to UME users who are stored in the LDAP directory server. UME group assignments are provisioned to those users that are stored in the database (for example, Administrator). LDAP group assignments are provisioned directly to the users that are stored in the directory server.

- Restrictions when connecting an AS Java to an LDAP Directory
 - Only the privilege assignments are provisioned to the AS Java repository. All other attributes are handled by the LDAP back-end repository. The user must exist in the LDAP repository before you assign any privileges to him or her in the Java repository. Assigning the Java account privilege or another Java role privilege to the user does not automatically create the user in the LDAP repository. You can still enable or disable users, because the hook for the back-end repository (LDAP) is defined.

10.2.5 Restrictions for LDAP Directory Connectors

Templates for the Sun Microsystems Sun One LDAP server and Microsoft Active Directory Server (ADS) are provided. You can adjust the tasks and jobs for other directory servers to meet your needs on a project base. Privilege grouping is not supported for LDAP directories.

10.2.6 Restrictions for SAP HANA Connectors

Validity

For the `MX_VALID_FROM` user attribute, take into account that, if the valid from date for the user is earlier than the current date, the user is provisioned to the SAP HANA system with a validity from the current date. This limitation comes from the SAP HANA system, in which you cannot create a user with a validity starting in the past.

Password Provisioning

If an administrator sets a productive password in the Identity Management system, the user also has to configure it in the SAP HANA system. For more information, see the SAP HANA Platform documentation.

Privilege Grouping

Since no multiple assignments are possible in the SAP HANA system, the SAP HANA Connector in the Identity Management system does not support assignment grouping. For this reason, privileges are provisioned to the SAP HANA system separately, but not as a group.

Setting Emails for Different Users

When you create a user and set its e-mail address with the `MX_MAIL_PRIMARY` attribute, you must use a unique e-mail. If you define an e-mail that already exists for another user, you will get an error message. That is a requirement of the SAP HANA system.

Creating Disabled Users in SAP HANA

The SAP HANA system does not support the creation of a disabled user in the backend. Due to a limitation in the SAP HANA system, you cannot post one SQL statement to directly create and at the same time deactivate a user.

Nevertheless, if you need to create a user that should be enabled after a certain period of time, you have the following options:

- Assign the SAP HANA account privilege (the so called ONLY privilege) to the user and configure the `valid_from` and `valid_to` properties of the assignment to meet your requirements.
- Create your custom process to create a disabled user in the SAP HANA system. This process should first create the user and then update the `MX_DISABLED` attribute value to the SAP HANA system.

Related Information

[SAP HANA Platform](#)

10.2.7 Restrictions for SuccessFactors Connector

Assignments

- Role assignments support just the user level (level 0) for both access and target populations.
- The Initial Load job does not load the user assignments from SuccessFactors.
- The assignment of a user to a permission group in SuccessFactors is not supported.

CRUD Operations and Provisioning

- Creating a user in IdM (using [CreateUser](#) process) and provisioning it to SuccessFactors is not supported.
- Deleting a user in IdM (using [DeleteUser](#) process) and provisioning the change to SuccessFactors is not supported.

- Modifying a user in IdM (using [ModifyUser](#) process) and provisioning the change to SuccessFactors is supported only for MX_HCM_SYSUNAME and MX_MAIL_PRIMARY user attributes.
- Enabling a user in IdM (using [EnableUser](#) process) and provisioning the change to SuccessFactors is not supported.
- Disabling a user in IdM (using [DisableUser](#) process) and provisioning the change to SuccessFactors is not supported.
- Setting a user password in IdM (using [SetUserPassword](#) process) and provisioning it to SuccessFactors is not supported.

Concurrent Employment

SAP Identity Management does not support Concurrent Employment. When it is used in Success Factors, all concurrent employments are loaded in SAP Identity Management as separate users.

Case-sensitive Usernames

In SuccessFactors, usernames can be configured as case-sensitive. This means that **Administrator** and **administrator** are treated like two separate users. However, the case-sensitive username attribute in SuccessFactors is mapped to the case-insensitive MSKEYVALUE attribute in SAP Identity Management.

Therefore, during initial load job **Administrator** and **administrator** are loaded as one user. The one that is loaded first, is created in SAP Identity Management. The one that is loaded second, modifies the already created user.

Last Modified On

When running the Initial Load job in delta mode, it is only possible to load changes up to three months into the past. This is due to a restriction in the Employee Central Compound Employee API that does not allow the `last_modified_on` filter parameter to be older than three months. For more information, see: [Thresholds and Limits of the CompoundEmployee API](#).

Any data changes older than this period will not be read and the job will result in an error. To load data changes older than that period, run the Reset Delta job followed by the Initial Load job, so that a full load is performed. For more information, see SAP Note [3117871](#) .

Contingent Workers

The Initial Load job does not load contingent workers (individuals who work for an organization on a non-permanent basis, such as contractors).

To load contingent workers, you need to add the SFSF_SOAP_FILTER parameter in the Read SFSF Employees (SOAP) pass and set the **isContingentWorker** filter to **true**. See [Parameters for FromSFSFSoap Pass \[page 1039\]](#)

Note that if this filter is added after you have run the Initial Load job, on your next run of the job not all contingent workers will be loaded. This is due to delta settings in the Write Delta Employee (SOAP) pass. To load all contingent workers, run the Reset Delta job followed by the Initial Load job, so that a full load is performed.

Related Information

[SAP SuccessFactors HCM Suite](#)
[Setting Up a SuccessFactors System \[page 640\]](#)

10.3 Considerations When Customizing the Provisioning Framework

If you need to modify the provisioning framework to meet your needs, then copy the corresponding templates to a custom folder and only modify the copied tasks. Make sure the links to tasks in repository definitions point to the copied tasks (as applicable).

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 an 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.

© 2025 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <https://www.sap.com/about/legal/trademark.html> for additional trademark information and notices.