SAP NetWeaver® Identity Management
Compliant provisioning using SAP Access Control

Configuration Guide
Preface

The product

The SAP NetWeaver Identity Management GRC integration consists of a set of tasks 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 NetWeaver 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 NetWeaver Identity Management performs risk validation on SAP Access Control before assigning permissions, in order to achieve the compliant provisioning.

The reader

This manual is intended for people who implement provisioning from SAP NetWeaver Identity Management 7.2 using SAP Access Control 10.0 (using the GRC provisioning framework).

Prerequisites

To get the most benefit from this manual, you should have the following knowledge and software:

- Knowledge of the Identity Center.
- Knowledge of the Virtual Directory Server.
- Knowledge of and access to SAP Access Control 10.0 SP4 or newer.
- SAP NetWeaver Identity Management Virtual Directory Server 7.2 SP5 or newer is correctly installed and licensed.
- SAP NetWeaver Identity Management Identity Center 7.2 SP5 or newer is correctly installed and licensed.
- The Provisioning Framework for SAP Systems is correctly installed and configured.

The manual

This document describes how you install and configure the GRC provisioning framework.
Related documents

You can find useful information in the following documents (all SAP NetWeaver Identity Management 7.2 relevant documentation is available on SAP Community Network, http://scn.sap.com/docs/DOC-8397):

- The install guides for the SAP NetWeaver Identity Management.
- *SAP NetWeaver Identity Management Compliant provisioning using SAP Access Control - Architectural overview.*
- *SAP NetWeaver Identity Management Identity Services Architectural overview.*
- *SAP NetWeaver Identity Management Identity Services Configuration guide.*
- The tutorials for the Identity Center.
- The tutorials for the Virtual Directory Server.
- Relevant documentation for SAP Access Control 10.0.
# Table of contents

**Introduction** ................................................................................................................................. 1  
Integration scenarios ......................................................................................................................... 2  
The configuration process .................................................................................................................. 4  
**Adding the Virtual Directory Server configuration** ................................................................. 5  
Setting the access credentials .............................................................................................................. 8  
Starting the server ............................................................................................................................... 8  
Testing the configuration .................................................................................................................... 9  
**Adding the GRC provisioning framework to the Identity Center** ............................................. 12  
Preparing the Identity Center .......................................................................................................... 12  
Importing the GRC provisioning framework .................................................................................... 13  
Importing the service jobs .................................................................................................................. 15  
Configuring the repository definition ............................................................................................... 16  
Configuring the Identity Service ........................................................................................................ 19  
**Process description** .................................................................................................................... 20  
**Initialization process description** .............................................................................................. 21  
Running the "Initial Load" jobs .......................................................................................................... 21  
Troubleshooting ................................................................................................................................. 26  
**Privilege assignment process description** ................................................................................... 27  
Distributed scenario ............................................................................................................................ 27  
Centralized scenario ........................................................................................................................... 28  
Grouping of privileges ....................................................................................................................... 29  
Obtaining the task values .................................................................................................................... 32  
**Task execution process description** ............................................................................................ 34  
AC Validation .................................................................................................................................. 34  
Fix PVO .......................................................................................................................................... 34  
Prepare AC Request .......................................................................................................................... 35  
Perform Risk Check ............................................................................................................................ 36  
Set IdM Approver ............................................................................................................................... 36  
Submit AC Request ............................................................................................................................ 36  
Write RequestId and opt. Start Polling ............................................................................................... 41  
Risk Validation ................................................................................................................................. 42  
AC Polling ........................................................................................................................................ 43  
AC Callback Service .......................................................................................................................... 45  
AC Validation - Risk Analysis Only .................................................................................................... 45  
**Typical use cases** ......................................................................................................................... 49  
Compliancy provisioning (role add and delete) for distributed scenario .......................................... 49  
Compliancy provisioning (role add and delete) for centralized scenario ......................................... 49  
Compliancy provisioning (role add, non-compliant removal) for centralized scenario .................... 50  
**Timeouts and stale requests** .......................................................................................................... 51  
**Limitations/tuning of the framework** ............................................................................................ 52  
**Configuring the Callback Service result handling scenario (AC Callback Service) on the GRC system/SAP Access Control** ......................................................................................... 53  
Activating GRC Web Services .......................................................................................................... 53  
Creating a connector ......................................................................................................................... 54  

© Copyright 2013 SAP AG. All rights reserved.
Maintaining connectors and connection types ................................................................. 54
Maintaining connection settings ...................................................................................... 54
Maintaining connector settings in Access Control ......................................................... 55
Maintaining mapping for actions and connector groups ............................................... 55
Introduction

This document gives an overview of the integration between SAP NetWeaver Identity Management 7.2 and SAP Access Control (for SAP Governance, Risk and Compliance (GRC) Solutions) 10.0.

The components of SAP NetWeaver Identity Management are used in the following way:

- **The 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.

- **The 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.
Integration scenarios

Several integration scenarios exist, depending on two factors:

- Landscape configuration
- Result handling

The GRC 10.0 Provisioning Framework can be configured to deal with any combination of these two.

Landscape configuration scenarios

There are two landscape configuration scenarios for the integration:

- Centralized provisioning: This is a scenario where SAP NetWeaver 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 NetWeaver Identity Management uses SAP Access Control to execute risk analysis. The centralized provisioning is recommended as a default solution.
- Distributed provisioning: This solution is recommended to use in exceptional cases only. This is a scenario where the provisioning is performed both by SAP NetWeaver Identity Management and SAP Access Control. SAP NetWeaver 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.

Result handling scenarios

Whenever a request to SAP Access Control is sent by the SAP NetWeaver 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:

- Polling: The Identity Management performs the appropriate web service request, polling the SAP Access Control for the result.
• 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, by executing its Exit service WS call to Identity Services.

The GRC 10.0 Provisioning Framework contains tasks/workflow that deals with both scenarios. The behavior of the provisioning framework is controlled by a configurable set of constants on the GRC10 repository definition as shown on page 16.

The configuration process

The configuration process described in this document consists of:
• Creating a configuration in the Virtual Directory Server based on a template.
• Importing the GRC 10.0 Provisioning Framework to the Identity Center.
• Configuring the imported objects.
• Optionally, configure the Identity Services if event based (AC Callback Service) result handling is needed.
Adding the Virtual Directory Server configuration

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.

It is assumed that you specified the location of the Java runtime environment, path to Keys.ini file, defined classpath/JDBC driver etc when you installed the Virtual Directory Server. The settings are defined in Tools/Options….

To create the configuration:

1. Choose File/New… to open the "New configuration" dialog box.

Select "GRC" in the "Group" list. Select "GRC AC 10.0 Integration.xml" in the "Template" list.
2. Choose "OK".

Fill in the following values (the names of the corresponding constants created in the Virtual Directory Server configuration are given in parentheses):

Port
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.

Web Service URL (URLPREFIX)
The URL to SAP Access Control system, typically with the pattern http://<server>:<port>/.

Client (URLCLIENT)
Specify the client for the web service.

GRC User and GRC Password (GRCUSER and GRCPWD)
Credentials of the user with access rights to execute web service calls against SAP Access Control.

Identity Center URL (CONNECTION_STRING)
Enter the connection string to the Identity Center database. It is recommended that you use the JDBC URL wizard. It is the <prefix>_rt user in the Identity Center database that should be used.

3. Choose "OK".

Enter a file name of the new configuration (for instance, grcintegration.xml) and save the configuration.
The expanded virtual tree looks like this:

The constants entered in the template above – URLPREFIX, GRCUSER, GRCPWD and CONNECTION_STRING can be altered. To do so, select the "Constants" node in the console tree of the Virtual Directory Server and view the properties:

![Constants Dialog Box](image)

Enter the changes and choose "Apply", then choose "OK" to close the "Constants" dialog box.
Setting the access credentials

The user access credentials to log on to the Virtual Directory Server are set to grcuser (user)/grcuser (password) as default in the template. It is not recommended to modify these, although they can be modified in the configuration:

1. Under the node "User groups" in the console tree, select the default authenticated user "grcuser" and view its properties.

   ![User properties - grcuser - Authenticated users](image)

   To change the user name enter the new login name in the "Login name" field.

2. Choose "Reset…” to change the password:

   ![Change password](image)

   Enter and confirm the new password.

3. Choose "OK" to close the "Change password" dialog box”.

4. Choose "Apply" to apply the changes, and then "OK” to close the user properties dialog box.

Starting the server

In order to verify that the server starts without errors, do the following:

1. Start the server.

2. Display the operation log (choose the "Operation" button).

3. Start the server. If the run-time environment is correct, the Virtual Directory Server will start listening on the configured port. 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. It can be changed by viewing the properties of Deployments/LDAP Deployments/Main listener.
Testing the configuration

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). It the sub-section below it is assumed that you have the LDP tool installed and available.

Logging in with LDP

1. Start LDP.
2. Choose Connection/Connect…:
   
   ![Connection/Connect.png]

   Enter the host name/IP number and port number you specified when configuring the Virtual Directory Server.

3. Choose "OK".
4. Choose Connection/Bind…:
   
   ![Connection/Bind.png]

   Enter grcuser (user)/grcuser (password) as user credentials to log on to the Virtual Directory Server. These are the default credentials in the template, but this can be modified in the configuration, as shown on page 8.

5. Choose "OK".
Performing a search

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". This corresponds to a node in the Virtual Directory Server configuration as shown on page 7.

1. Choose **Browse/Search**.

   ![Image of Search dialog box]

   Make sure that the "Attributes" field is empty.

2. Choose "Options":

   ![Image of Search Options dialog box]

   Make sure that the "Attributes" field is empty.

3. Choose "OK" to close the "Search Options" dialog box.
4. Choose "Run" to perform the search.

The applications returned by the search may vary depending on what is available in SAP Access Control you are connecting to.
Adding the GRC provisioning framework to the Identity Center

The GRC provisioning framework added to the Identity Center makes it possible to submit requests to SAP Access Control from a provisioning solution implemented in the Identity Center.

The framework files are available with the SAP NetWeaver Identity Management installation. The necessary files are:

- **GRC 100 Provisioning Framework.mcc** (the framework itself), located in directory `<Identity Center install directory>\Templates\Identity Center\SAP Provisioning framework`.
- Three service jobs, **AC 10.0 - Initial Load - Commons.dst**, **AC 10.0 - Initial Load - Centralized provisioning.dst** and **AC 10.0 - Initial Load - Distributed provisioning.dst**, located in directory `<Identity Center install directory>\Templates\Identity Center\Jobs\GRC`.

Adding the provisioning framework involves the following steps:

- Preparing the Identity Center
- Importing the GRC 10.0 Provisioning Framework
- Importing the service jobs
- Configuring operation
  - Configuring the repository information
  - Performing Initial Load

Preparing the Identity Center

Before importing the framework to the identity store, some initial configuration of the Identity Center needs to be made.

- Ensure that you have at least one valid dispatcher enabled to run Java jobs.

Specifying import options

To specify import options:

1. View the properties of the Identity Center and select the "Options" tab.
2. Make sure that "Enable imported jobs" is selected
3. Select a default dispatcher for the imported jobs.
   
   This ensures that imported tasks/jobs are enabled. It is possible to enable those later, but this will have to be done manually for each job in the framework and the number of jobs is large.
4. Choose "Apply".
Importing the GRC provisioning framework

The GRC 10.0 Provisioning Framework contains tasks specific to the SAP Access Control 10.0 integration with SAP NetWeaver Identity Management.

To import the framework:

1. Select the identity store where you will import the framework and choose "Import…" from the context menu.
2. Locate the file containing the framework, *GRC 100 Provisioning Framework.mcc*.
3. Choose "Open".

Make sure that "Import" is selected.
4. Select the "Advanced" tab and make sure that a dispatcher is selected for the imported jobs, as configured for the Identity Center.
5. Choose "Next >".
6. Choose "Import".
7. Choose "Finish" when the import is completed. Alternatively choose "View logfile" before choosing "Finish" to view the details about the completed import.
The imported framework is added to the identity store:

- A sub-folder **Utility tasks** which contains a number of utility tasks.
- An ordered task group **AC Validation** (for more information see the section **AC Validation** on page 34).
- An ordered task group **AC Polling** (for more information see the section **AC Polling** on page 43).
- An ordered task group **AC Callback Service** (for more information see the section **AC Callback Service** on page 45).
- An ordered task group **AC Validation - Risk Analysis Only** (for more information see the section **AC Validation - Risk Analysis Only** on page 45).

In addition, when imported the framework adds:

- a set of scripts
- a set of variables and constants needed to control the framework behavior
- a repository definition **GRC10**. For more information about the repository definition and its constants, see the section **Configuring the repository definition** on page 16.
Importing the service jobs

The service jobs are used for the initial load and other tasks that are not part of the provisioning framework itself. Three files are available:

- AC 10.0 - Initial Load - Commons.dst
- AC 10.0 - Initial Load - Centralized provisioning.dst
- AC 10.0 - Initial Load - Distributed provisioning.dst

Which one you need to use depends on which landscape configuration scenario you want to deploy.

To import the jobs, do the following:

1. Select the job folder where you want to add the service jobs. You can either use an existing folder or create one for this purpose.

   **Note:**
   *Name of the job folder may be "<Name of your identity store> - <Name of the repository definition> jobs", e.g. "Enterprise People - GRC jobs".*

   Choose **New/Run job wizard…** from the context menu.

2. Choose "Next >" and then locate the initial load job template, AC 10.0 – Initial Load – Commons (file AC 10.0 - Initial Load - Commons.dst). Choose "Change folder…" to browse for the correct folder.

3. Choose "Next >".

4. Choose "Next >" and then "Finish" to complete the wizard.

5. Repeat the procedure above for the initial load jobs AC 10.0 – Initial Load – Distributed provisioning and AC 10.0 – Initial Load – Centralized provisioning.
Configuring the repository definition

When importing the GRC 10.0 Provisioning Framework a repository definition named \textit{GRC10} is created (it may be necessary to refresh the repository node in the Identity Center Management Console).

View the repository constants for the \textit{GRC10} repository definition and modify the constants.

**Virtual Directory Server related constants**

These constants define the values needed for successful communication with the Virtual Directory Server.

The following constants have hard-coded values and \textbf{should not} be changed. They can however be changed, but the appropriate values in the template have to be changed accordingly:

<table>
<thead>
<tr>
<th>Constant name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDS2GRC_SUFFIX</td>
<td>The top RDN of the virtual tree in the Virtual Directory Server. The default value is &quot;o=grc&quot;.</td>
</tr>
<tr>
<td>VDS2GRC_BRANCH_LOOKUPSERVICE</td>
<td>The RDNs of the branches in the virtual tree of the Virtual Directory Server (one branch for each of the exposed web services).</td>
</tr>
<tr>
<td>VDS2GRC_BRANCH_REQUESTSTATUS</td>
<td></td>
</tr>
<tr>
<td>VDS2GRC_BRANCH_RISKANALYSIS</td>
<td></td>
</tr>
<tr>
<td>VDS2GRC_BRANCH_ROLEDETAILS</td>
<td></td>
</tr>
<tr>
<td>VDS2GRC_BRANCH_SEARCHROLES</td>
<td></td>
</tr>
<tr>
<td>VDS2GRC_BRANCH_SELECTAPPLICATIONS</td>
<td></td>
</tr>
<tr>
<td>VDS2GRC_BRANCH_USERACCESSREQUEST</td>
<td></td>
</tr>
<tr>
<td>VDS2GRC_BRANCH_PROVISIONINGLOG</td>
<td></td>
</tr>
<tr>
<td>VDS2GRC_BRANCH_REQUESTDETAILS</td>
<td></td>
</tr>
<tr>
<td>VDS2GRC_BRANCH_EXITSERVICE</td>
<td></td>
</tr>
<tr>
<td>VDS2GRC_BRANCH_EUPCONFIG</td>
<td></td>
</tr>
</tbody>
</table>

The following values must be configured. Use the values specified when configuring the Virtual Directory Server:

<table>
<thead>
<tr>
<th>Constant name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDS2GRC_HOST</td>
<td>The host name or IP address of the Virtual Directory Server.</td>
</tr>
<tr>
<td>VDS2GRC_PORT</td>
<td>The LDAP port which is used by the Virtual Directory Server.</td>
</tr>
<tr>
<td>VDS2GRC_LOGIN/ VDS2GRC_PASSWORD</td>
<td>The credentials of the user that is used to log on to the Virtual Directory Server. The default value is \textit{grcuser/grcuser} (user/password). These default credentials in the template can be modified in the configuration, as shown on page 8.</td>
</tr>
</tbody>
</table>
## Request constants

These constants will be a part of the request sent to SAP Access Control (the web service call `SAPGRC_AC_IDM_USERACCESSREQUEST`):

<table>
<thead>
<tr>
<th>Constant name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRC_MANAGER_ID</td>
<td>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 (hence it must have approval access rights in SAP Access Control). The default value is NULL. See the section <em>Setting the manager ID</em> on page 37 in this document.</td>
</tr>
<tr>
<td>GRC_REASON</td>
<td>The default reason for the request. It is used only if the reason is not set by other means (see more about this later in the document on page 39).</td>
</tr>
<tr>
<td>GRC_COMPANY</td>
<td>The default company string. Make sure that this company exists in your Access Control configuration. Whether the GRC_COMPANY attribute is mandatory for a request or not depends on your Access Control configuration.</td>
</tr>
<tr>
<td>GRC_REQUESTOR_ID/ GRC_REQUESTOR_EMAILADDRESS/ GRC_REQUESTOR_INITSYSTEM</td>
<td>Correct values of one of the valid users in the SAP Application Server Java which runs SAP Access Control. These values are used in case it is not possible to obtain them from the user entry that executed the role assignment task. For more information, see the section <em>Setting the requestor properties</em> on page 38 in this document.</td>
</tr>
</tbody>
</table>
Run-time constants

These constants set the values that are necessary for the internal processing of the provisioning framework.

<table>
<thead>
<tr>
<th>Constant name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MX_PRIV_GROUPING_RULE</td>
<td>P:0 (do not change). Only relevant for distributed provisioning scenario. For more, see section Grouping of privileges on page 29.</td>
</tr>
<tr>
<td>MX_PRIV_GROUPING_ATTRIBUTE</td>
<td>The name of the attribute whose value may be used for grouping, i.e. in addition to other criteria, only the privileges that have values identical to the value of this attribute will be grouped. Only relevant for distributed provisioning scenario.</td>
</tr>
<tr>
<td>REPOSITORY_TYPE</td>
<td>GRC (do not change)</td>
</tr>
<tr>
<td>MX_ADD_MEMBER_TASK/MX_DEL_MEMBER_TASK</td>
<td>The constants hold the task ID of the task AC Validation from the GRC 10.0 Provisioning Framework. The values are filled in automatically during the import of the GRC provisioning framework. These attribute are only relevant for the distributed provisioning scenario. For the centralized scenario they will be set to &quot;-1&quot;, meaning that no add/delete member tasks are inherited and executed even if defined on the repository definition.</td>
</tr>
<tr>
<td>MX_VALIDATE_ADD_TASK/MX_VALIDATE_DEL_TASK</td>
<td>The constants hold the task ID of the task AC Validation from the GRC 10.0 Provisioning Framework. The values are by default set to &quot;-1&quot;, meaning that no tasks are inherited and executed. These attributes are only relevant for the centralized provisioning scenario where the GRC provisioning framework is used for validation.</td>
</tr>
<tr>
<td>MX_AC_POLLING_ENABLED</td>
<td>If set to value &quot;1&quot;, then the GRC 10.0 Provisioning Framework will after successfully sending a request to Access Control invoke the polling process in order to obtain the result of the request. The task referenced by the attribute MX_AC_POLLING_TASK is executed.</td>
</tr>
<tr>
<td>MX_AC_POLLING_TASK</td>
<td>Holds the task to be executed during the polling process. See above.</td>
</tr>
<tr>
<td>MX_AC_FIX_SYSTEM_PRIVILEGES</td>
<td>Legal values are:</td>
</tr>
<tr>
<td></td>
<td>• 0: Required for the centralized provisioning scenario, i.e. keeping track of the accounts in the target systems is the responsibility of SAP Provisioning Framework.</td>
</tr>
<tr>
<td></td>
<td>• 1: Required for the distributed provisioning scenario, i.e. the GRC 10.0 Provisioning Framework keeps track of accounts in various target systems.</td>
</tr>
</tbody>
</table>
Adding the GRC provisioning framework to the Identity Center
SAP NetWeaver Identity Management Compliant provisioning using SAP Access Control Configuration Guide

<table>
<thead>
<tr>
<th>Constant name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MX_AC_APPROVAL_SYSTEMACCOUNT</td>
<td>Holds the MSKEYVALUE of the user that is responsible for approvals generated by this framework. This user must not have any UI permissions for the Identity Management User Interface in the Identity Center. Will be created during the Initial Load(s). Do not change.</td>
</tr>
</tbody>
</table>

### Configuring the Identity Service

When implementing the event-based result handling scenario (AC Callback Service, available only for SAP Access Control 10.0 as mentioned on page 3), the following parameters must be configured 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:

![Configuring the Identity Service](image)

Use the same values as specified in the Identity Center:

<table>
<thead>
<tr>
<th>Constant name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROLE_ASSIGNMENT_CALLBACK_TASK</td>
<td>Holds the task ID of the task AC Callback Service from the GRC 10.0 Provisioning Framework. See more details about the task AC Callback Service on page 45.</td>
</tr>
<tr>
<td>ROLE_DEFINITION_CALLBACK_TASK</td>
<td>For future use.</td>
</tr>
</tbody>
</table>

For details about deploying to 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).
Process description

In the following sections, three phases and their processes are described:

- Initialization
- Privilege assignment
- Task execution
Initialization process description

During this 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.

This is achieved by running one of the "Initial Load" jobs. The section below describes this process.

Running the "Initial Load" jobs

Before any provisioning or de-provisioning through the provisioning framework can be carried out, the framework must be initialized.

The following job always has to be executed:

- AC 10.0 – Initial Load – Commons

In addition, one of the following two jobs has to be executed:

- AC 10.0 – Initial Load – Centralized provisioning
- AC 10.0 – Initial Load – Distributed provisioning

Which one of the two jobs should be executed depends on which landscape configuration scenario (see page 2) is chosen.

The centralized provisioning scenario requires the AS ABAP – Initial Load job (from the SAP Provisioning Framework) to be executed first, before the job AC 10.0 – Initial Load – Centralized provisioning is executed. For the distributed provisioning scenario, only the job AC 10.0 – Initial Load – Distributed provisioning is required to be executed.

Initial load job – common for both scenarios

Disregarding of what provisioning scenario is to be implemented, the job AC 10.0 – Initial Load – Commons must be executed.

- AC 10.0 – Initial Load – Commons
  - System log
  - Job log
  - Scripts
  - Job variables
  - Job constants
  - Create AC System Account
  - Get Priorities From VDS
  - Create Priorities Variables
  - Get Request Types From VDS
  - Create Request Types Variables
  - Get Employee Types From VDS
  - Create Employee Types Variables
  - Get Role Action Types From VDS
  - Create Role Action Variables
  - Get Mandatory Parameters for Submit AC Request From VDS
  - Create list of Mandatory Parameters for Submit AC Request
  - Get Applications From VDS
  - Get Roles From VDS

© Copyright 2013 SAP AG. All rights reserved.
It contains the following passes:

- **Create AC System Account**: Creates an account that is needed for the approval process triggered by the framework. For this purpose the MSKEYVALUE held by the attribute MX_AC_APPROVAL_SYSTEMACCOUNT on the repository definition GRC10 is used.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSKEYVALUE</td>
<td>sapMX_AC_APPROVAL_SYSTEMACCOUNT%</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>System Account for External Approvals in Access Control</td>
</tr>
</tbody>
</table>

- **Get Priorities From VDS**: The information about the priorities is obtained from the Virtual Directory Server using the web service call `LookupService`. 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 global (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 `LookupService`. 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 global (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 `LookupService`. 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 global (employee) variables. A typical example is `GRC_ROLE_EMPLOYEE_PERMANENT` with value "002".

- **Get Role Action Types From VDS**: The information about the action types is obtained from the Virtual Directory Server using the web service call `LookupService`. The information is stored in the local table `sapGRC10roleactiontypes`.

- **Create Role Action Variables**: Uses the script `sap_processActionCode` to process the information stored in the local table `sapGRC10roleactiontypes` and construct the global (role action) variables. A typical example is `GRC_ROLE_ACTION_ASSIGN` with value "006".

- **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 `EUPConfig`. The information is stored in the local table `sapGRC10eupconfig`.

- **Create List of Mandatory Parameters for Submit AC Request**: Uses the script `sap_processEUPConfigResult` to process information stored in the local table `sapGRC10eupconfig` and create the global 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 `SelectApplications`. The information is stored in the local table `sapGRC10applications`.

- **Get Roles From VDS**: For each application, the information about roles available in SAP Access Control is obtained using the web service call `SearchRoles`. The information is stored in the local table `sapGRC10roles`.
The global variables available after running this job may be the following:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRC_ROLE_ACTION_ASSIGN</td>
<td>006</td>
</tr>
<tr>
<td>GRC_ROLE_ACTION_Become</td>
<td>009</td>
</tr>
<tr>
<td>GRC_ROLE_ACTION_RETAIN</td>
<td>010</td>
</tr>
<tr>
<td>GRC_ROLE_EMPLOYEE_PERMANENT_LONG</td>
<td>001</td>
</tr>
<tr>
<td>GRC_ROLE_EMPLOYEE_TEMPORARY_LONG</td>
<td>002</td>
</tr>
<tr>
<td>GRC_ROLE_PRIORITY_SP_HI_LONG</td>
<td>002</td>
</tr>
<tr>
<td>GRC_ROLE_PRIORITY_SP_LOW_LONG</td>
<td>003</td>
</tr>
<tr>
<td>GRC_ROLE_PRIORITY_SP_MID_LONG</td>
<td>004</td>
</tr>
<tr>
<td>GRC_ROLE_REQUEST_ACCOUNT_DELETIONS</td>
<td>005</td>
</tr>
<tr>
<td>GRC_ROLE_REQUEST_CHANGES_TO_ANEXISTINGACCOUNT</td>
<td>001</td>
</tr>
<tr>
<td>GRC_ROLE_REQUEST_INFORMATION</td>
<td>009</td>
</tr>
<tr>
<td>GRC_ROLE_REQUEST_LOCKING_ACCOUNTS</td>
<td>010</td>
</tr>
<tr>
<td>GRC_ROLE_REQUEST_NEW_ACCOUNT</td>
<td>011</td>
</tr>
<tr>
<td>GRC_ROLE_REQUEST_NEW_HIRE</td>
<td>012</td>
</tr>
<tr>
<td>GRC_ROLE_REQUEST_SP_REQ_CHANGE_USER</td>
<td>015</td>
</tr>
<tr>
<td>GRC_ROLE_REQUEST_SP_REQ_CREATE_USER_DESC</td>
<td>016</td>
</tr>
<tr>
<td>GRC_ROLE_REQUEST_SP_REQ_DELETE_USER_DESC</td>
<td>017</td>
</tr>
<tr>
<td>GRC_ROLE_REQUEST_SP_TST_REQ_1_DESC_LONG</td>
<td>018</td>
</tr>
<tr>
<td>GRC_ROLE_REQUEST_UNLOCKING_ACCOUNTS</td>
<td>021</td>
</tr>
<tr>
<td>GRC_SUBMIT_AC_REQUEST_MANDATORY_PARAMETER_1</td>
<td>REQUEST_TYPE</td>
</tr>
<tr>
<td>GRC_SUBMIT_AC_REQUEST_MANDATORY_PARAMETER_2</td>
<td>REQUESTEMAILADDRESS</td>
</tr>
<tr>
<td>GRC_SUBMIT_AC_REQUEST_MANDATORY_PARAMETER_3</td>
<td>REQUESTREASON</td>
</tr>
<tr>
<td>GRC_SUBMIT_AC_REQUEST_MANDATORY_PARAMETER_4</td>
<td>BPROC</td>
</tr>
<tr>
<td>GRC_SUBMIT_AC_REQUEST_MANDATORY_PARAMETER_5</td>
<td>FIRSTNAME</td>
</tr>
<tr>
<td>GRC_SUBMIT_AC_REQUEST_MANDATORY_PARAMETER_6</td>
<td>LASTNAME</td>
</tr>
<tr>
<td>GRC_SUBMIT_AC_REQUEST_MANDATORY_PARAMETER_7</td>
<td>EMPJOB (DefwebDEV)</td>
</tr>
<tr>
<td>GRC_SUBMIT_AC_REQUEST_MANDATORY_PARAMETER_8</td>
<td>EMAILADDRESS</td>
</tr>
</tbody>
</table>

**Initial load job for centralized provisioning scenario**

For the centralized provisioning scenario, the following jobs are required to be executed in the following order:

- **AS ABAP – Initial Load** job from the SAP Provisioning Framework.
- **AC 10.0 – Initial Load – Centralized provisioning** imported from the GRC 10.0 Provisioning Framework.

Perform the initial load **AS ABAP – Initial Load** as described in the document SAP NetWeaver Identity Management Identity Management for SAP System Landscape: Configuration guide (see section 3.6 Performing the Initial Loads).
The imported AC 10.0 – Initial Load – Centralized provisioning job contains the following passes:

- **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:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS_KEYVALUE</td>
<td>$FUNCTION sap_matchACRoleToIdMPrivilege|application|name|role%</td>
</tr>
<tr>
<td>MX_APPLICATION_ID</td>
<td>%application%</td>
</tr>
<tr>
<td>MX_AC_ROLEID</td>
<td>%name%</td>
</tr>
<tr>
<td>MX_AC_ROLETYPE</td>
<td>%role%</td>
</tr>
<tr>
<td>DESCRIPTION</td>
<td>%description%</td>
</tr>
<tr>
<td>ADD_TASK REFERENCES</td>
<td>AC manages only a subset of roles</td>
</tr>
<tr>
<td>MX_ADD_MEMBER_TASK</td>
<td></td>
</tr>
<tr>
<td>MX_DEL_MEMBER_TASK</td>
<td></td>
</tr>
<tr>
<td>MX_REPOSTRY_VALIDATE</td>
<td>GROE</td>
</tr>
</tbody>
</table>

The script sap_matchACRoleToIdMPrivilege is used to match the SAP Access Control roles to the SAP NetWeaver 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 definition that holds a constant
  AC_APPLICATION_ID with a value that matches MX_APPLICATION_ID. Then try to find privilege named
  PRIV:ROLE:<name of just obtained repository definition>:<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.
Initial load job for distributed provisioning scenario

For the distributed provisioning scenario, the AC 10.0 – Initial Load – Distributed provisioning job is required to be executed.

This job contains the following passes:

- **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.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAMESPACE</td>
<td>PRIN%</td>
</tr>
<tr>
<td>APPLICATION_ID</td>
<td>%application%</td>
</tr>
<tr>
<td>ROLE_ID</td>
<td>%name%</td>
</tr>
<tr>
<td>ROLE_TYPE</td>
<td>%type%</td>
</tr>
<tr>
<td>Repository_Name</td>
<td>%repo_Name%</td>
</tr>
<tr>
<td>mx_Privilege_Type</td>
<td>GRC</td>
</tr>
</tbody>
</table>

- **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 `sapGRC10roles (%application%, %name%, etc).
  - Name of the repository definition.
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 definition 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.

Example 1:
Privilege could not be uniquely identified – an ABAP role exists with the same name as a profile:

| Messages | Error | AC Role Z_DEV (DC3800) found as ROLE and PROFILE ... Skip |

The error indicates that a role object in Access Control couldn't be mapped to an ABAP role privilege or an ABAP profile privilege in the Identity Management. The reason for this is that both a role and a profile exist with that name.

A solution to this is to manually perform the assignment maintaining the following attributes through e.g. a task in User Interface:

- MX_AC_ROLEID: must contain the role name as in the Access Control system.
- MX_APPLICATION_ID: must contain the same application name as in the Access Control system, of the application to which the object belongs to.
- MX_AC_ROLETYPE: must contain the type of the role as defined in the Access Control system.

Example 2:
Privilege not found in the identity store:

| Messages | Error | Could not map AC Role PLY_GROUP:PORTAL:VC GROUP (IWIDM) to IDM Privilege ... Skip |

This error indicates that the role object loaded from the Access Control system is not present in the identity store. This can be solved by running either an initial load job or an update job for the AS ABAP system.

- 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 de-select 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.
Privilege assignment process description

In this section, the privilege assignment/de-assignment process is described for the distributed and the centralized scenario.

Distributed 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 attribute MX_AC_FIX_SYSTEM_PRIVILEGES, on the repository definition GRC10, must be set to value "1" for the distributed provisioning scenario.

The following describes the process that takes place when a set of privileges is assigned/de-assigned to/from a certain user entry:

- The administrator assigns to/de-assigns from a user a set of roles/privileges, or an end-user requests a role/privilege assignment/de-assignment.

  *Note: Privileges that correspond to 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.*

- For these privileges, the Identity Management obtains the value of add member event task (or remove member event task, depending on the operation). See the section Obtaining the task values on page 32 for information about how this is achieved. If the value is defined, the Identity Management creates a pending object for each of the assigned (removed) privileges.

- The Identity Management checks the property "Grouping rule" (MX_PRIV_GROUPING_RULE) on the repository definition which the privileges belong to (GRC10), and if necessary groups the pending objects. For more information, see the section Grouping of privileges on page 29.

- Only one pending object in the group will actually trigger the provisioning process. The GRC 10.0 Provisioning Framework is designed to process all pending objects in a group, and will create a single request to SAP Access Control where all assigned privileges are sent.
Centralized scenario

In this section, the privilege assignment/de-assignment process for centralized scenario is described.

The existing privileges on ABAP repository definitions, enriched (by adding properties necessary for operations towards GRC) during the Initial Load phase, are configured so that any assignment/de-assignment 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/de-assignment: the privilege is not assigned to/removed from an entry until the execution operation is successfully executed.

The attribute MX_AC_FIX_SYSTEM_PRIVILEGES, on the repository definition GRC10, must be set to value "0" for the centralized provisioning scenario.

The following describes the process that takes place when a set of privileges is assigned to/de-assigned from a certain user entry:

- The administrator assigns to/de-assigns from a user a set of roles/privileges, or an end-user requests a role/privilege assignment/de-assignment.

  **Note:** Privileges that correspond to 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.

- For these privileges, the Identity Management obtains the value of the task MX_VALIDATE_ADD/DEL_TASK (see the section Obtaining the task values on page 32 for information about how this is achieved).
  - 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.

- The Identity Management checks the property "Grouping rule", i.e. the attribute MX_PRIV_GROUPING_RULE, of the repository definition which is defined for validation (here, GRC10). If necessary, the pending objects are grouped. For more information, see the section Grouping of privileges on page 29.

- Only one pending object in the group will actually trigger the validation process. The GRC 10.0 Provisioning Framework is designed to process all pending objects in a group, and will create a single request to SAP Access Control where all assigned privileges are sent.

- If validation is successful, the Identity Management continues the provisioning procedure, checking and executing the tasks MX_ADD/DEL_MEMBER_TASK defined for privileges.
Grouping of privileges

Grouping of the privileges is done according to the grouping policy defined by the property "Grouping rule" (MX_PRIV_GROUPING_RULE) on the repository definition. Possible settings for this property are shown below:

<table>
<thead>
<tr>
<th>Grouping policy/rule</th>
<th>Split on</th>
<th>Back-end system</th>
<th>Operation type (add/remove)</th>
<th>Privilege attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>No grouping. Every pending object will trigger the GRC provisioning framework and will be sent in its own request to SAP Access Control.</td>
</tr>
<tr>
<td>P:7</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Same operations sent to the same system and having the same specific privilege attribute defined are sent in one request.</td>
</tr>
<tr>
<td>P:6</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>All pending objects sent to the same system with a specific privilege attribute defined are grouped together into one request.</td>
</tr>
<tr>
<td>P:5</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Pending objects with same operations and a specific privilege attribute defined are sent in the same request.</td>
</tr>
<tr>
<td>P:4</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>All pending objects with a specific privilege attribute defined are grouped and sent in the same request.</td>
</tr>
<tr>
<td>P:3</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>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.</td>
</tr>
<tr>
<td>P:2</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>All operations to the same system/application will be sent in the same request.</td>
</tr>
<tr>
<td>P:1</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Same operations, regardless of the system/application, will be grouped together and sent in the same request.</td>
</tr>
<tr>
<td>P:0</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>All pending objects, regardless of the operation and the system, are sent in the same request.</td>
</tr>
</tbody>
</table>
The 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.
Grouping across repository definitions

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 repository definitions). 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 definition 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 repository definitions 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 repository definitions (master grouping rule, used if different grouping rules are defined for each repository definition).

**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 repository definitions each with their own set of privileges.
- Each of these ABAP repository definitions 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 task AC Validation, which actually is an ordered task group).
- A set of privileges is assigned to a user, where the privileges belong to different repository definitions.
- The privilege assignment requests are expected to be grouped into one single request, across the ABAP repository definitions.
According to the basic concept of grouping, where only requests belonging to the same repository definitions are grouped, grouping across the repository definitions would fail as we here have requests from several ABAP repository definitions. 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 task AC Validation. When this task is the same for all ABAP repository definitions, as is the case here, then it is possible to group requests from multiple repository definitions 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 repository definitions but with the same MX_OPERATION_TASKID value can be grouped together (i.e. even if they don't belong to the same repository definitions).

The task AC Validation is suitable for grouping across repository definitions and is defined as such, and the master grouping rule can be optionally defined. This is defined on the "Result handling" tab of the ordered task group AC Validation, under section "Custom parameters":

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MX_TASK_GROUPABLE</td>
<td>Sling</td>
<td>1</td>
</tr>
<tr>
<td>MX_TASK_GROUPING_RULE</td>
<td>Sling</td>
<td>P:0</td>
</tr>
</tbody>
</table>

Define the following:

- MX_TASK_GROUPABLE: with value 1 (TRUE)
- MX_TASK_GROUPING_RULE: with values P:0 through P:7 (optional)

**Obtaining the task values**

When performing provisioning/de-provisioning with GRC provisioning framework, the following task- and repository definition 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

The following sections describe how the task values are obtained.
Obtaining values for tasks given by MX_ADD/DEL_MEMBER_TASK

For the tasks 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 not defined on the privilege, then from the repository definition defined by the attribute MX_REPOSITORY_ADD/DEL_MEMBER.
- If MX_REPOSITORY_ADD/DEL_MEMBER not defined, then from repository definition defined by the attribute MX_REPOSITORYNAME.

For examples, see section Typical use cases on page 45.

Obtaining values for tasks given by MX_VALIDATE_ADD/DEL_TASK

For the tasks 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 definition defined by the attribute MX_REPOSITORY_VALIDATE.
- If MX_REPOSITORY_ADD/DEL_MEMBER not defined, then from repository definition defined by the attribute MX_REPOSITORYNAME.

For examples, see section Typical use cases on page 45.
Task execution process description

This section describes the execution process of the tasks in the GRC provisioning framework.

AC Validation

Upon the assignment/de-assignment of the SAP Access Control privileges, the Identity Management executes the defined add member event task/remove member event task.

The task AC Validation performs both risk analysis and the provisioning, and its use as such is intended for the distributed scenario. The task can also be used in centralized scenario, where a few adjustments on both the GRC10 repository definition and on GRC side need to be done in order for the task to be used in centralized scenario. The task AC Validation - Risk Analysis Only described on page 45 is preferred to use in centralized scenario.

The task AC Validation is an ordered task group:

- Fix PVO
- Prepare AC Request
- Perform Risk Check

The task group consists of three tasks:

- Fix PVO
- Prepare AC Request
- Perform Risk Check

It also configures specific chain result actions:

<table>
<thead>
<tr>
<th>Chain result actions</th>
<th>Execute task on OK result</th>
<th>Execute task on Failed result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1843/Reapply System Privileges if needed</td>
<td>1844/Skip Pending On Group</td>
</tr>
</tbody>
</table>

If the task is successfully executed, as the last step the system will recalculate the necessary system privileges that the entry should have.

If any of the tasks in the chain results in error, the task Skip Pending On Group will be executed. It uses the script sap_skipPendingOnGroup and removes all pending objects in the particular pending object group without applying them to the user.

Fix PVO

This task is a part of the task group AC Validation and calls a script 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.
Prepare AC Request

This task is a part of the task group AC Validation and uses the script `sap_grc10_prepareRiskCheckExecution` to:

- 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.

The following list of context variables may be of interest when creating custom scripts as mentioned on page 38:

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRCROLEIDLIST</td>
<td>The list of roles approved by SAP Access Control. Format: role-id@system, ...</td>
</tr>
<tr>
<td>PENDINGMSKEY</td>
<td>The MSKEY of the pending object (the leading in the group).</td>
</tr>
<tr>
<td>PENDINGUSERNAME</td>
<td>The MSKEYVALUE and the MSKEY of the user entry that the privilege(s) is/are assigned to.</td>
</tr>
<tr>
<td>PENDINGPRIVILEGEREPNAME</td>
<td>The repository definition defined for the privilege.</td>
</tr>
<tr>
<td>PENDINGPRIVILEGEMSKEY</td>
<td>The MSKEY of the privilege(s) assigned.</td>
</tr>
<tr>
<td>GROUP_PENDINGMSKEY_PRIVILEGEMSKEYS</td>
<td>The list of all MSKEYs in the group.</td>
</tr>
<tr>
<td>GROUP_DISTINCT_APPLICATIONS</td>
<td>The list of all application IDs, for all privileges in the group.</td>
</tr>
<tr>
<td>GROUP_APPLICATIONS</td>
<td>Context variables used by the scripts in GRC 10.0 Provisioning Framework.</td>
</tr>
<tr>
<td>USEDSYSTEMAPPLID</td>
<td>For the Callback Service result handling scenario (AC Callback Service).</td>
</tr>
<tr>
<td>GROUP_SYSTEM_PRIVILEGES</td>
<td>For the Callback Service result handling scenario (AC Callback Service).</td>
</tr>
<tr>
<td>CLEAN_SYSTEM_PRIVILEGES</td>
<td>For the Callback Service result handling scenario (AC Callback Service).</td>
</tr>
<tr>
<td>GROUP_SYSTEM_PRIVILEGE_IDS_AND_NAMES</td>
<td>For the Callback Service result handling scenario (AC Callback Service).</td>
</tr>
<tr>
<td>MX_AC_OVERALLSTATUS</td>
<td>For the Callback Service result handling scenario (AC Callback Service).</td>
</tr>
<tr>
<td>MX_AC_OVERALLREASON</td>
<td>For the Callback Service result handling scenario (AC Callback Service).</td>
</tr>
</tbody>
</table>
Perform Risk Check

The task *Perform Risk Check* is a part of a task group *AC Validation*. After a successful execution of the task *Prepare AC Request*, the framework has all information needed to start the process of constructing an AC request.

This is performed by an ordered task group, *Perform Risk Check*:

- **Set IdM Approver**
  The task *Set IdM Approver* is a part of the ordered task group *Perform Risk Check*. The pass sets the approver for the task:

  - The approver set for the task is just a placeholder for the approval process in the SAP Access Control system. The system account has no Identity Management User Interface permissions in the Identity Center.

- **Submit AC Request**
  This is a major task in the task group *Perform Risk Check* and it is responsible for preparing the information that is sent to Virtual Directory Server (that in turn executes a `SAPGRC_AC_IDM_USERACCESSREQUEST` 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`.
In the section below, it is explained how some of the important parameters that are submitted in the request are calculated and filled in. The typical Identity Management pass that sends the information to the Virtual Directory Server, which in turn will create request to SAP Access Control, looks like this:

### Common parameters
- **DN**: Identifier in the console (virtual) tree. The parameter triggers the creation and sending of the web service call `SAPGRC_AC_IDM_USERACCESSREQUEST`.
- **changetype**: Internal attribute, used for internal processing in the Virtual Directory Server.

### Internal attributes for the Virtual Directory Server
- **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 `UserAccessRequest` 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 definition of the GRC 10.0 Provisioning Framework defines the default manager ID (`GRC_MANAGER_ID`) that may be used by the task mentioned above (optionally NULL):

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID: GRC_COMPANY</td>
<td>SAP</td>
</tr>
<tr>
<td>ID: GRC_MANAGER_ID</td>
<td>TESTUSER</td>
</tr>
<tr>
<td>ID: GRC_REASON</td>
<td>Sent by Netweaver IDM : User need...</td>
</tr>
</tbody>
</table>
The GRC 10.0 Provisioning Framework uses the following algorithms to set the values for this parameter:

- **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 definition `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 (SAPGRC_AC_IDM_USERACCESSREQUEST) have following three properties:

- **REQUEST_TYPE**
- **REQUEST_PRIORITY**
- **REQUEST_EMPLOYEETYPE**

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>`) is 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. See the list of some of the relevant context variables on page 35.

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 global variables (see global variables on page 23). The abovementioned custom scripts can, if desired, use these variables.

**Setting the requestor properties**

The web service `Submit User Access Request` requires requestor ID (and optionally the requestor properties) – the identity of the user that performed the role assignment.

The repository definition of the GRC provisioning framework defines the following default requestor properties that may be used by the task mentioned above:

- **GRC_REQUESTOR_EMAILADDRESS**
- **GRC_REQUESTOR_FIRSTNAME**
- **GRC_REQUESTOR_ID**

The GRC provisioning framework uses the following algorithm to set the values for the parameters **REQUESTORID** and **REQUESTOREMAILADDRESS**: 
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 definition.

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_initsystem%) is obtained manually in the configuration process.

Setting the reason
The web service Submit Request has optional reason parameter. The setting of this parameter is performed using the script sap_grc10_setRequestReason.

Script sap_grc10_setRequestReason
This script takes one parameter – the default reason configured on the repository definition GRC10. If any of the assigned privileges have an assignment reason configured, the script will retrieve these and create a single reason string that is returned and used in the request sent to SAP Access Control.

If this is not a case, a default reason (the script parameter) will be returned (see page 17).

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 global variables (see global variables on page 23).

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:

<table>
<thead>
<tr>
<th>Mandatory attributes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FIRSTNAME</td>
<td>SRC_FIRSTNAME</td>
</tr>
<tr>
<td>LASTNAME</td>
<td>SRC_LASTNAME</td>
</tr>
<tr>
<td>EMAILADDRESS</td>
<td>SRC_EMAILADDRESS</td>
</tr>
<tr>
<td>TELEPHONE</td>
<td>SRC_TELEPHONE</td>
</tr>
</tbody>
</table>

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.

Parameter 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:

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.

**Write RequestId and opt. Start Polling**

The task *Write RequestId and opt. Start Polling* is a part of the ordered task group *Perform Risk Check*.

The pass calls the script `sap_grc10_WriteRequestId2PVO` to retrieve the request ID from a context variable (MX_GRC_REQUEST_ID as of, and GRC_REQUEST_ID prior to SAP NW Identity Management 7.1 SP4) and save it to MX_AC_REQUESTID on a pending value object. It then checks the attribute MX_AC_POLLING_ENABLED of the assigned privilege (defined on the repository definition 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.
Risk Validation

The Risk Validation is an approval task and is a part of the ordered task group Perform Risk Check.

- **Approve**
  - Fix System Privileges
  - Ordered Group
    - Remove System Application Privileges
    - Apply Pending On Group - approved only
  - Skip Pending On Group - approved only

- **Decline**
  - Skip Pending On Group
  - Set Context Variable for Checking Task Failure
  - Check Task Failure
  - True
    - Skip Task
  - False
    - Skip Pending On Group

When you configure an approval task, you use the "Approval" tab to configure whether the task requires an approval and who is to grant this approval. To do this you specify one or more approval rules defining who is responsible for handling approval requests.

The approval task has two sub-nodes, "Approve" and "Decline". The tasks under the "Approve" node are executed when the request is approved, and the tasks under the "Decline" node are executed if the request is declined:

**Approve:**
The task Apply Pending On Group – approved only is executed if the request is approved. It retrieves the value of the attribute MX_AC_RESULT from a pending value object (obtained either by polling (AC Polling) or a callback service (AC Callback Service) as a result handling scenario), and applies the approved privileges only.

Depending on the setting of the repository definition 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).

**Decline:**
There are two possible reasons why the "Decline" sub-node is executed:

- The request has been declined (a valid action performed by the approver).
- The entire process of obtaining the assignment result fails, e.g. because of a timeout or some other reason (error).

The task performs a check on which one of the two reasons mentioned above has appeared. The task Skip Pending On Group is executed in both cases:

- If the request is declined, then the Skip Pending On Group is the only task executed. The AC Validation task ends with status "OK" and the final result of the assignment is "REJECTED".
- If obtaining of the assignment result fails, the overall result of the AC Validation task is set to error. Then the chain error task defined on the AC Validation is executed (task Skip Pending On Group), and the result of the assignment is set to "FAILED".
AC Polling

This is an ordered group of tasks, containing two tasks:

- **Read Status**
- **Check Status**

**Read Status**

This task polls SAP Access Control for result using the value of the context variable `MX_GRC_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. Possible values are "OK" and "FAILED".

**Check Status**

This task will process the status obtained by the task **Read Status**. Possible values are "FAILED", "OK" and "ELSE". A request can either be approved or declined, or the approval may fail (e.g. because of the timeout). Task status is processed in the following way:

- In case of status "FAILED", which means that the request was declined by an approver, the task **DeclineRequest** will be executed.
- In case of status "OK", it is possible that only a subset of the roles sent to SAP Access Control are 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 task **Write Role Result to PVO** stores the contents of the context variables `GRCSTATUS` and `GRCROLEIDLIST` to a pending value object. After the request is approved (task **ApproveRequest**), the GRC provisioning framework will apply only the privileges (from the pending object group) that are approved and not apply all others.
• In case of status "ELSE", which means that the approval failed, the task TaskFailed is executed. The task sets a failure flag on the pending value object (task Set TaskFailure Flag using the script sap_grc_WriteTaskFailureFlag2PVO) and executes the DeclineRequest task.

The behavior is configured on the "Result handling" tab of the AC Polling task – the task TaskFailed is referenced by "Execute task on Failed result":

**Timeouts**

In this section aligning of the timeout settings for the approval task Risk Validation with the retry interval of the Read Status pass is described. The goal is to configure a good error handling in case the request does not get approved within a defined period of time. In order to achieve this, the approval timeout should match the retry interval setting. The following default values are configured in the GRC provisioning framework:

• The approval task Risk Validation has a default timeout set to 5 days:

• The required retry interval of the Read Status pass is then defined to be 1440 retries, every 5 minutes:

In addition, the adequate settings on the SAP Access Control side are needed. See section Timeouts and stale requests on page 51, for more.
AC Callback Service

This is an ordered group of tasks, containing two tasks:

- Prepare Callback Service execution
- External Result

Prepare Callback Service

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 GRC's Exit service WS call to Identity Services.

The Identity Services processes the information in the request, saves the necessary information in the MX_AC_RESULT variable and starts the execution of the task AC Callback Service.

The job Set Context Variables uses the global script sap_grc10_PrepareExitServiceExecution to read the information (the request status) from MX_AC_RESULT, process it and save the created result to the context variable MX_AC_OVERALLSTATUS.

External Result

This task will process the status obtained by the task Prepare Callback Service (the status saved to the context variable MX_AC_OVERALLSTATUS). The status is processed in the following way:

- In case of status "REJECTED", which means that the request was declined (rejected) by the approver, the task DeclineRequest is executed.

- In case of anything else but the status "REJECTED", the request is approved (task ApproveRequest is executed).

AC Validation - Risk Analysis Only

The task AC Validation - Risk Analysis Only is an ordered task group:

- Fix PVO
- Prepare AC Request
- Perform Risk Analysis
- Check Risk Return Status

This task group consists of four tasks, described in the following subsections:

- Fix PVO
- Prepare AC Request
- Perform Risk Analysis
- Check Risk Return Status
The task group has some similarities to the previously described task AC Validation. The difference between the tasks AC Validation and AC Validation - Risk Analysis Only is that the former performs both risk analysis and provisioning, but the latter performs only risk analysis. Therefore, the intended usage of the task AC Validation - Risk Analysis Only is the centralized provisioning scenario. It is still possible to use the task AC Validation for the centralized scenario in exactly same way as before.

Where it was required to change some parameters and flags both on the GRC10 repository definition and on the GRC side in order for the AC Validation task to be used for centralized scenario, this is not the case for the task AC Validation - Risk Analysis Only. The only thing that needs to be done is to configure the task as the member event task ("Validate add task") on the repository definition GRC10:

<table>
<thead>
<tr>
<th>General</th>
<th>Privilege</th>
<th>Event tasks</th>
<th>Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>Privilege member event tasks:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Validate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Validate add task:</td>
<td>101851/AC Validation - Risk Analysis Only</td>
<td></td>
<td>When valid</td>
</tr>
<tr>
<td>Validate remove task:</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Validate modify validity:</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assignment</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Add task:</td>
<td>None</td>
<td></td>
<td>When valid</td>
</tr>
<tr>
<td>Remove task:</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modify validity task:</td>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Fix PVO**

This is the same task as for the task group AC Validation described on page 34.

**Prepare AC Request**

This is the same task as for the task group AC Validation described on page 35.

**Perform Risk Analysis/Submit Risk Analysis Request**

The task Perform Risk Analysis is a part of the ordered task group AC Validation - Risk Analysis Only. 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:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DN</td>
<td>cn=MSKEYVALUE43%2BrepVD0526RC_BRANCH_RISKANALYSISIS%2BrepVD0526RC_SURF</td>
</tr>
<tr>
<td>changetype</td>
<td>4BD</td>
</tr>
<tr>
<td>AUDITID</td>
<td>$FUNCTION: sap_grc10_cmAuditor$</td>
</tr>
<tr>
<td>OBJECTID</td>
<td>2MSKEYVALUE44</td>
</tr>
<tr>
<td>OBJECTTYPE</td>
<td>USR</td>
</tr>
<tr>
<td>ROLEDATA</td>
<td>$FUNCTION: sap_grc10_getRoleData(FUNCTION:sap_grc10_getContextVar [GROUP_PENDINGMSKEYS_PRIVILEGESCRYSTALS])$</td>
</tr>
<tr>
<td>SIMULATIONOBJTYPE</td>
<td>ROL</td>
</tr>
<tr>
<td>SIMULATIONRISKONLY</td>
<td>X</td>
</tr>
<tr>
<td>ADDLATTRIB</td>
<td>06,05,14</td>
</tr>
<tr>
<td>APPLICATIONTYPE</td>
<td></td>
</tr>
<tr>
<td>BUSINESSPROC</td>
<td></td>
</tr>
<tr>
<td>HITCOUNT</td>
<td></td>
</tr>
<tr>
<td>ORGLEVEL</td>
<td></td>
</tr>
<tr>
<td>ORGRLUE</td>
<td>value1,value2</td>
</tr>
<tr>
<td>ORGVAL</td>
<td>value1,value2</td>
</tr>
<tr>
<td>REPORTFORMAT</td>
<td></td>
</tr>
<tr>
<td>REPORTTYPE</td>
<td>value1,value2</td>
</tr>
<tr>
<td>RISKLEVEL</td>
<td></td>
</tr>
<tr>
<td>RICLESID</td>
<td></td>
</tr>
<tr>
<td>ROLESETID</td>
<td></td>
</tr>
<tr>
<td>USERGROUP</td>
<td></td>
</tr>
<tr>
<td>USERTYPE</td>
<td></td>
</tr>
</tbody>
</table>

**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 Virtual Directory Server**

- **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.

**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 (,) as seen in the screen shot above (e.g. 06,05,14 for the attribute ADDLATTRIB).
Check Risk Return Status

This switch task has three sub-nodes (states):

- **ERROR**: an error occurred when processing the risk analysis and no status is obtainable.
- **OK**: the risk analysis is successfully performed.
- **Else**: something unpredictable has happened (i.e. something went wrong) when processing the risk analysis and no status is obtainable.

The same task is executed in case of ERROR or Else, the task *Fail Pending On Group*. This task will set the status of the task execution (of the task *AC Validation - Risk Analysis Only*) to FAILED. The roles and privileges will also have the status FAILED.

When the risk analysis is successfully performed (status OK), the risks (if any) are logged in the Job log and the roles/privileges with the risk are added to the pending value object (the tasks *Print Risks* and *Write Risk Result to PVO* respectively). Then the task *Apply Pending On Group* – based on risks is executed, which will either apply the role/privilege if no risks are discovered (and thus the subsequent provisioning tasks will be executed) or decline the role/privilege if any risks are discovered (no provisioning will then take place, i.e. the subsequent tasks will not run).
Typical use cases

This section describes three typical use cases for the GRC 10.0 Provisioning Framework and their setup:

- Compliancy provisioning (role add and delete) for distributed scenario
- Compliancy provisioning (role add and delete) for centralized scenario
- Compliancy provisioning (role add and non-compliant delete) for centralized scenario

Compliancy provisioning (role add and delete) for distributed scenario

There are two possible setups for this use case:

- Setup 1:
  - Define the task reference attributes MX_ADD_MEMBER_TASK and MX_DEL_MEMBER_TASK on the GRC repository definition (not ABAP).
  - Define the repository definition reference attribute MX_REPOSITORYNAME on the privileges to be the GRC repository definition.
- Setup 2:
  - Define both the task reference attributes MX_ADD_MEMBER_TASK and MX_DEL_MEMBER_TASK, and the repository definition reference attribute MX_REPOSITORYNAME (GRC, and not ABAP, repository definition) on the privileges.

Compliancy provisioning (role add and delete) for centralized scenario

There are two possible setups for this use case:

- Setup 1:
  - Define the task reference attributes MX_ADD_MEMBER_TASK and MX_DEL_MEMBER_TASK on the ABAP repository definition.
  - Define the task reference attributes MX_VALIDATE_ADD_TASK and MX_VALIDATE_DEL_TASK on the GRC repository definition (not ABAP).
  - On privileges, define the attributes MX_REPOSITORYNAME and MX_REPOSITORY_VALIDATE to hold to references to ABAP and GRC repository definitions respectively.
- Setup 2:
  - ABAP repository definition: None.
  - GRC repository definition: None.
  - On privileges, define the following attributes:
    - MX_REPOSITORYNAME: ABAP repository definition.
    - MX_REPOSITORY_VALIDATE: GRC repository definition.
Typical use cases

SAP NetWeaver Identity Management Compliant provisioning using SAP Access Control Configuration Guide

- **MX_ADD_MEMBER_TASK**: Use the add member task from the provisioning framework for SAP systems.
- **MX_DEL_MEMBER_TASK**: Use the delete member task from the provisioning framework for SAP systems.
- **MX_VALIDATE_ADD_TASK**: Use the validate add task from the GRC provisioning framework.
- **MX_VALIDATE_DEL_TASK**: Use the validate delete task from the GRC provisioning framework.

**Compliance provisioning (role add, non-compliant removal) for centralized scenario**

There are three possible setups for this use case:

**Setup 1:**
- Define the attribute **MX_DEL_MEMBER_TASK** on the ABAP repository definition.
- Define the attribute **MX_ADD_MEMBER_TASK** on the GRC repository definition.
- On privileges, define the following attributes:
  - **MX_REPOSITORYNAME**: ABAP repository definition
  - **MX_REPOSITORY_ADD_MEMBER**: GRC repository definition

**Setup 2:**
- Define the attribute **MX_DEL_MEMBER_TASK** on the ABAP repository definition.
- Define the attribute **MX_ADD_MEMBER_TASK** on the GRC repository definition.
- On privileges, define the following attributes:
  - **MX_REPOSITORYNAME**: GRC repository definition
  - **MX_REPOSITORY_ADD_MEMBER**: ABAP repository definition

**Setup 3:**
- ABAP repository: None.
- GRC repository: None.
- On privileges, define the following attributes:
  - **MX_REPOSITORYNAME**: ABAP repository definition
  - **MX_REPOSITORY_ADD_MEMBER**: GRC repository definition
  - **MX_DEL_MEMBER_TASK**: Use the delete member task from the provisioning framework for SAP systems.
  - **MX_VALIDATE_ADD_TASK**: Use the validate add task from the GRC provisioning framework.
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 shown in section *Timeouts* on page 44.

For more information about the configuration of the stale requests, see the documentation for SAP Access Control 10.0.
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 `SAPGRC_AC_IDM_USERACCESSREQUEST` 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 section *Mandatory attributes* on page 39.

- The `SAPGRC_AC_IDM_USERACCESSREQUEST` web service call includes the following information: Request type, priority and employee type. The values that SAP NetWeaver Identity Management uses when executing this call has to be aligned with the values that are configured in SAP Access Control.

  See section *Setting the requestor properties* on page 38.

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.
Configuring the Callback Service result handling scenario (AC Callback Service) on the GRC system/SAP Access Control

To configure the Callback Service on the GRC system/SAP Access Control, you need to complete the following steps in the following order:

1. Activating GRC Web Services
2. Creating a connector (for the Callback Service result handling scenario)
3. Maintaining connectors and connection types
4. Maintaining connection settings
5. Maintaining connector settings in Access Control
6. Maintaining mappings for actions and connector groups

There is a separate section for each of the steps describing the details. The first five steps are generic configuration steps for the integration between SAP NetWeaver 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".

Activating GRC Web Services

In this section, we generate the GRC Service Providers to activate the GRC Web Services. This will allow SAP NetWeaver Identity Management to access the services via GRC Web Service calls:

1. Navigate to Common Component Settings/Integration Framework in the structure of your GRC system.
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 available on http://scn.sap.com/docs/DOC-26208.

Note:
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.
Creating a connector

A connector, which can be used for Callback Service and a HTTP connection, needs to be created. Do the following:

1. Navigate to **Common Component Settings/Integration Framework** in the structure of your GRC system.
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 (e.g. /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.

Maintaining connectors and connection types

As previous section defined the HTTP connection for the connector, the steps in this section will 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:

1. Navigate to **Common Component Settings/Integration Framework** in the structure of your GRC system.
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.

Maintaining connection settings

You need to assign the connector to an integration scenario. Do the following:

1. Navigate to **Common Component Settings/Integration Framework** in the structure of your GRC system.
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".*

Maintaining connector settings in Access Control

You need to create the connector and maintain the connector settings for the system connected to the access control application. Do the following:

1. Navigate to **Governance, Risk and Compliance/Access Control** in the structure of your GRC system.
2. Start the IMG activity "Maintain Connector Settings".
3. Follow the IMG activity documentation to create connector "NW_IDM72" with application type "11".

Maintaining mapping for actions and connector groups

You need to assign the actions to the connector group and select the default connector for the group. Do the following:

1. Navigate to **Governance, Risk and Compliance/Access Control** in the structure of your GRC system.
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:
6. Assign group parameter mapping, and especially make sure that the parameter "EXIT_FROM_GRC" is set to "TRUE":

7. Save the entries before exiting the IMG activity.