SAP NetWeaver® Identity Management
Identity Center

Tutorial
- Provisioning

Version 7.0 Rev 2
Preface

The product
SAP NetWeaver Identity Management Identity Center is the primary component for identity management. The Identity Center includes functions for identity provisioning, workflow, password management, logging and reporting. It uses a centralized repository, called the identity store, to provide a uniformed view of the data, regardless of the data's original source.

The reader
This manual is written for people who need an introduction to the provisioning in the Identity Center.

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

• Knowledge of LDAP.
• Knowledge of Microsoft SQL Server or Oracle.
• General knowledge about the Identity Center and job definitions, for instance as described in the SAP NetWeaver Identity Management Identity Center Tutorial: Basic synchronization and SAP NetWeaver Identity Management Identity Center Getting Started.

The following software is required:

• SAP NetWeaver Identity Management Identity Center 7.0 SP2 or newer, correctly installed and licensed.
• A directory server with the external object classes top, person and inetOrgPerson (as defined in RFC 2798). The credentials necessary to add, modify and delete entries in the directory server are also required.
• An Identity Center where at least one dispatcher has been configured and is running. (See SAP NetWeaver Identity Management Identity Center Getting Started.)
• An LDAP client to view the contents of the directory.

The manual
This tutorial consists of six sections containing information about how you build a task structure and run a provisioning system.

This tutorial is not a substitution for training.

Person names used in this tutorial are fictional.
Related documents

You can find useful information in the following documents:

- SAP NetWeaver Identity Management Identity Center Getting Started
- SAP NetWeaver Identity Management Identity Center Tutorial: Basic synchronization
Table of contents

Introduction ........................................................................................................................................ 1
   The repositories ........................................................................................................................... 1
   The data flow and task structure ............................................................................................... 2
   Preparations .............................................................................................................................. 2
   Section overview ....................................................................................................................... 4
Section 1: Building the identity store ............................................................................................... 5
   Defining a repository definition for the hr.csv file .................................................................... 5
   Creating the identity store ......................................................................................................... 7
   Reading the HR data into the identity store ............................................................................ 9
   Verifying the contents of the identity store ............................................................................. 15
   Enabling the delta ..................................................................................................................... 17
Section 2: Preparing the repositories ............................................................................................... 19
   Adding a repository definition for the LDAP server ............................................................... 19
   Defining additional repository constants .............................................................................. 21
   Creating the organization ......................................................................................................... 23
   Adding a repository definition for the file system folder ........................................................ 26
Section 3: Adding the create LDAP user tasks ............................................................................. 28
   Creating a folder for the LDAP tasks ........................................................................................ 28
   Adding task: #LDAP_Provisioning ......................................................................................... 29
   Adding task: Create LDAP entry ............................................................................................. 30
   Testing the tasks ....................................................................................................................... 34
   Adding task: #LDAP_Modify ................................................................................................... 37
Section 4: Adding the LDAP privilege ............................................................................................ 40
   Creating the LDAP privilege .................................................................................................... 40
   Removing the repository from the top-level task .................................................................... 42
   Configuring the Workflow interface ....................................................................................... 43
   Hiding the "Provisioning folder" ............................................................................................. 44
   Creating the folder .................................................................................................................... 44
   Creating the workflow task ....................................................................................................... 45
   Adding the privilege to the identity store entry ....................................................................... 48
Section 5: Adding the delete LDAP user tasks ............................................................................. 55
   Adding task: #LDAP_Deprovisioning ...................................................................................... 55
   Adding task: Disable LDAP entry ........................................................................................... 56
   Adding task: Delete LDAP entry ............................................................................................. 58
   Testing the tasks ....................................................................................................................... 60
   Updating the LDAP privilege .................................................................................................. 61
   Removing the repository definition from the task ................................................................... 62
Section 6: Resetting the tutorial data ............................................................................................. 63
   Resetting the directory server ................................................................................................. 63
   Emptying the identity store ...................................................................................................... 64
   Resetting the delta information ............................................................................................... 66
   Running the job ......................................................................................................................... 67
Section 7: Automatically assigning the privilege ................................................................. 68
  Updating the "HR to identity store" job .................................................................................. 68
  Running the job ..................................................................................................................... 68

Section 8: Adding the file system tasks ............................................................................... 69
  Adding task: #FILE_Provisioning ......................................................................................... 69
  Adding sub-task: Add file ..................................................................................................... 70
  Testing the tasks .................................................................................................................. 70
  Adding task: #FILE_Modify ................................................................................................. 72
  Adding task: #FILE_Deprovisioning ..................................................................................... 74
  Testing the tasks .................................................................................................................. 76
  Updating the "Reset tutorial data" job .................................................................................. 76

Section 9: Adding the FILE privilege .................................................................................. 78
  Creating the FILE privilege ................................................................................................. 78
  Removing the repository reference from the tasks ............................................................... 79
  Assigning the FILE privilege to the entries ...................................................................... 80
  Running the provisioning system ....................................................................................... 82

Section 10: Handling deleted entries .................................................................................... 83
  Disabling deleted entries ...................................................................................................... 83
  Updating the "Reset tutorial data" job .................................................................................. 85
  Testing the mechanism ....................................................................................................... 86

Section 11: Adding the telephone numbers ........................................................................ 87
  Adding a repository definition for the tel.csv file .............................................................. 87
  Adding the telephone numbers ........................................................................................... 88
  Updating the identity store ................................................................................................. 91
  Updating the "#LDAP_Modify" task ..................................................................................... 92
  Running the job .................................................................................................................. 93
  Enabling the delta ............................................................................................................... 93
  Removing deleted attributes ............................................................................................... 95
  Updating the "Reset tutorial data" job .................................................................................. 97
  Running the provisioning system ....................................................................................... 97
  What's next? ....................................................................................................................... 98
Introduction

The purpose of this tutorial is to show how to configure the Identity Center and build a task structure for provisioning and de-provisioning of employees. This involves building an identity store where information about all employees is stored, defining the different data sources as repositories in the Identity Center, and defining the tasks and jobs that perform the provisioning. You will also see how to reset the identity store and provisioning system for testing purposes.

The repositories

For the sake of simplicity, this tutorial has as few external dependencies as possible. The purpose is to show how to create an identity data flow, more than showing how to connect to a number of different repository types.

For this reason, the repositories are ASCII files, with the exception of a directory server, which can also be replaced by ASCII files.

The ASCII repositories can be replaced by other types of repositories to achieve a more realistic scenario. The From ASCII file passes that are used to read the ASCII files must then be replaced by a pass type suitable to read the type of data source in question.

The following figure gives an overview of the repositories involved in this tutorial:

Below is a description of the repositories:

<table>
<thead>
<tr>
<th>Repository</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hr.csv</td>
<td>An ASCII file in CSV format containing entries from an HR system.</td>
</tr>
<tr>
<td>tel.csv</td>
<td>An ASCII file in CSV format containing telephone numbers.</td>
</tr>
<tr>
<td>Directory Server</td>
<td>The users will be written to this directory server as part of the provisioning process.</td>
</tr>
<tr>
<td>File system</td>
<td>This is a folder in the file system where the contents of the identity store are stored with one file per user.</td>
</tr>
</tbody>
</table>

The files are installed as part of the product installation and can be found in the sub-directory \tutorial\data source of the Identity Center installation directory.

The tutorial shows how data is read from the files hr.csv and tel.csv into the identity store. From there, tasks in the provisioning system are used to update the target repositories.
The data flow and task structure

The following diagram illustrates the data flow that we are going to implement in this tutorial:

There are two jobs that read the data from the repositories and update the entries in the identity store. The entry type for these entries is MX_PERSON.

We create two privileges (LDAP and FILE) that we assign to the entries. The privileges contain links to tasks that are executed when the privilege is assigned or removed, or when an entry that has the privilege assigned is modified.

The task structure is shown in the illustration above. There are separate task structures for each of the target repositories.

Preparations

Before you proceed with the tutorial, there are a couple of things that must be specified.

Defining a global constant

To be able to reference the files that are used in this tutorial in a uniform way, we create a global constant containing the path to the directory where the files are stored:

1. Select the "Global constants" entry in the console tree and choose New/Constant... from the context menu (right-click the entry to open the context menu):

   ![Global constant dialog box]

   Specify the name of the constant and the directory where the data source files are stored.

2. Choose "OK" to close the dialog box.
Specifying the system log level

To be able to view the log information shown in this tutorial, you must make sure that the log level for the system log is set to "Info". If necessary, change the log level and choose "Apply".
## Section overview

The tutorial consists of the following sections:

<table>
<thead>
<tr>
<th>Section 1: Building the identity store</th>
<th>In this section we are going to create the identity store and read the contents of the file hr.csv into the identity store.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Section 2: Preparing the repositories</td>
<td>This section describes how you prepare and define the target repositories.</td>
</tr>
<tr>
<td>Section 3: Adding the create LDAP user tasks</td>
<td>Here, you see how you build the task structure for the tasks that updates the LDAP server.</td>
</tr>
<tr>
<td>Section 4: Adding the LDAP privilege</td>
<td>In this section we will add the privilege. We will also add the workflow task that is used to add the privilege to the entries in the identity store.</td>
</tr>
<tr>
<td>Section 5: Adding the delete LDAP user tasks</td>
<td>This section describes how we add the tasks for deleting a user in the directory server. We will also update the privilege to contain a deprovisioning task link.</td>
</tr>
<tr>
<td>Section 6: Resetting the tutorial data</td>
<td>The section describes how we create a job that resets all the generated data.</td>
</tr>
<tr>
<td>Section 7: Automatically assigning the privilege</td>
<td>In this section we will see how we can automatically assign the privilege when the users are added to the identity store.</td>
</tr>
<tr>
<td>Section 8: Adding the file system tasks</td>
<td>This section shows how to add the tasks for updating the file system.</td>
</tr>
<tr>
<td>Section 9: Adding the FILE privilege</td>
<td>In this section we will add a privilege for the file system folder.</td>
</tr>
<tr>
<td>Section 10: Handling deleted entries</td>
<td>This section describes how you handle entries that are deleted from the master data source.</td>
</tr>
<tr>
<td>Section 11: Adding the telephone numbers</td>
<td>The last section describes how you add the telephone numbers and e-mail addresses from the second data source.</td>
</tr>
</tbody>
</table>
Section 1: Building the identity store

In this section we are going to create the identity store for the entries and read the contents of the file hr.csv into the identity store.

Defining a repository definition for the hr.csv file

A repository definition is used to hold constants and variables which are common for one data source (repository). The repository constants can be accessed from the context menu in the same way as global constants.

1. Start the repository wizard by selecting the "Repositories" entry in the console tree, and choosing New/Repository… from the context menu.
2. Choose "Next >".
3. Navigate to the "Repositories" sub-directory and select "File".
4. Choose "Next >".

[Diagram of repository wizard]

Repository wizard - Select a template

Repository wizard - Repository name
Enter a name and description of the repository.

4. Choose "Next >".

Use the context menu to insert the global constant we added and add the name of the file (hr.csv).

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

The repository is added to the console tree:
Creating the identity store

The Identity Center supports many logical identity stores. When installing the Identity Center, there is a default identity store. We are going to define a new identity store, "Tutorial".

1. Select the entry "Identity stores" and choose **New/Identity store...** from the context menu to start the identity store wizard and choose "Next >".

![Identity store wizard](image)

Enter a name for the identity store.

2. Choose "Next >".

![Identity store wizard](image)

We will use the MX_PERSON entry type, so we do not need any additional entry types.

3. Choose "Next >" and then "Finish".
The identity store is added to the console tree:
Reading the HR data into the identity store

We now have created a repository definition for the hr.csv file and defined an identity store that we can use when creating the job.

Creating the folder and job

First, we are going to create a folder for the jobs in the tutorial, and the job definition for this job.

1. Create a folder called "Identity Center tutorial" that can be used to hold the jobs. Select the Identity Center's entry in the console tree and choose New/Folder… from the context menu to create the folder.

2. Create a job by selecting the folder's entry and choosing New/Empty job from the context menu.

Modify the name of the job in the console tree.

Enable the job and select a dispatcher.

3. Choose "Apply".

This job will contain two passes; one to read the ASCII file (hr.csv) into the temporary table (tutorial_HR), and another to read from this table into the identity store. This must be done in a single job. The reason is that the first pass will delete the temporary table every time it executes, and then fill it with the data from the hr.csv file. If the second pass was a separate job (which could then be run asynchronously from the first), it could start just when the table was deleted or just partly filled, and then remove the missing people from the identity store.
Reading the ASCII file

First, we will create the pass that reads the ASCII file:

1. Select the job in the console tree and choose **New/From ASCII file** from the context menu.

   ![Console tree screenshot]

Enter *Read HR* as the name of the pass in the console tree.

**Repository**

Select the "Tutorial-HR" in the "Repository" list.
2. Select the "Source" tab:

File name
Use the context menu to insert the repository constant `%$rep.FILENAME%` that refers to the file name.

Field separator
Enter a comma sign (,) as the field separator.

Header line
Make sure that "Header line" is selected.
3. Select the "Destination" tab:

Fill in the fields with the following values:

**Database**
Use the context menu to insert the system parameter %$ddm.identitycenter% that refers the Identity Center database.

**Table name**
Enter TUTORIAL_HR as the table name.

*Note:*
*Do not use hyphen in table names, as this will cause problems with some database drivers.*

**Definitions**
Choose "Insert template" and select "Data source template" to create the pass definitions.

4. Choose "Apply".

**Running the job**
At this point, we are ready to test the pass. Run the job by viewing the job properties and choosing "Run now". View the job log to verify that the job ran successfully, and that a number of entries have been processed.
Updating the identity store

The next step is to create the pass that writes the data to the identity store:

1. Select the "Read HR" pass and choose **New/To Identity store** from the context menu and select the "Source" tab:

   ![Identity store image](image)

   Modify the pass name in the console tree.

   **Database**
   Use the context menu to insert the system parameter `%Sddm.identitycenter%`.

   **SQL statement**
   Enter the SQL statement to select all rows from the table created in the previous pass.
2. Select the "Destination" tab:

- **Identity store**
  Select the "Tutorial" identity store.

- **Entry type**
  Select the entry type "MX_PERSON".

- **Definitions**
  Choose "Insert template" and select "Data source template" to insert the definitions for the pass. Modify the definition to use the attributes from the entry type. You can use the context menu to find the destination attributes.

3. Choose "Apply".

**Testing the job**

Run the job and open the job log to verify that 50 entries were added (100 entries processed).
Verifying the contents of the identity store

If everything has gone well, the identity store should now contain all entries from the hr.csv file.

**Note:**
Make sure that the Monitoring web interface is configured for the Identity Center you are using.

1. Start the Monitoring web interface.

2. Choose "Identity store" in the menu.
3. Select the "Tutorial" identity store and then "Search" to return all entries in the identity store.

4. Verify that the entries are present in the identity store.
Enabling the delta

We now have two working passes. The next step is to ensure that only modified entries in the data source are written to the identity store. The delta mechanism must be enabled on the "To Identity store" pass of the "HR to identity store" job.

1. Select the "HR to ID store" pass and select the "Delta" tab:

![Image of the interface for enabling delta]

Fill in the fields with the following values:

**Enable delta**
Select this check box to enable delta on this pass.

**Delta database**
Use the context menu to insert the system parameter %Sddm.identitycenter% to specify that you want to use the Identity Center database for the delta database.

**Delta identifier**
Enter TutorialHR as the delta identifier. This must be unique within one delta database.

**Delta key**
This is automatically filled in with the value from the first line of the definitions on the "Destination" tab.

2. Choose "Apply".
Run the job a couple of times and view the job log:

The first time the job is run after the delta is enabled, 50 entries are modified, while the next time, the job detects that the entries are unmodified.

**Note:**
*The count is the total for the job, including the entries handled by the "Read HR" pass. These entries are always included in the "Add" column, as no delta has been defined for this pass.*
Section 2: Preparing the repositories

We are going to update two target repositories:

- An LDAP server meeting the requirements described in the preface.
- A file system folder. This is used to show an alternative type of repository. One file will be created for each user, containing information about the user.

Both these repositories will be added as repository definitions in the Identity Center. The repository definitions are then referenced from the jobs and tasks we create.

Adding a repository definition for the LDAP server

Before we can provision the entries to the LDAP server, we must add a repository definition for the LDAP server and create the organization where we will add the entries.

To add a repository definition for the LDAP server:

1. Start the repository wizard by selecting the "Repositories" entry in the console tree, and choosing **New/Repository** from the context menu.
2. Choose "Next >".

Navigate to the "Repositories" sub-directory and choose "Directory".
3. Choose "Next >".

Enter *Tutorial-LDAP* as the name of the repository definition.

4. Choose "Next >".

Fill in information about your directory server. The user must have access to create an organization below the specified starting point.

5. Choose "Next >" and then "Finish".
The repository definition is added to the console tree with the following repository constants:

![Repository Constants](image)

### Defining additional repository constants

In addition to the constants created by the wizard, we need to create some additional repository constants:

- The name of the organization.
- The DN of the organization.
- The DN of the entries within the organization.

### Adding a repository constant for the organization

To add the constant:

1. Select the repository's "Constants" node in the console tree and choose **New/Constant…** from the context menu:

   ![Repository Constant Dialog](image)

   This constant contains the name of the organization you are going to add to the directory server.

2. Choose "OK".
Adding the repository constant for the organization DN

Add the DN constant in the same way as the previous constant with the following values:

Use the context menu to insert the repository constants.

Adding the repository constant for the entries' DN

Add the constant for the DN of the entries that we will add to the organization:

You have to enter the first part manually, as the source attributes are not available from the context menu at this point.
Creating the organization

Before we can run the tasks, we must create the organization in the directory server.

This pass will be part of a job that is used to reset the tutorial data; the identity store, repositories and delta. This is often necessary during a development phase. In addition, this will verify access to the directory server.

1. Select the folder "Identity Center tutorial" in the console tree and choose New/Empty job from the context menu.

Rename the job in the console tree to Reset tutorial data.

Modify the job properties:

- **Enabled**
  Select this check box to enable the job.

- **Run by dispatchers**
  Select a dispatcher that is responsible for running this job.

2. Choose "Apply".
3. Create the pass by selecting the job and choosing **New/Run pass wizard** to start the pass wizard.

4. Choose "Next >".

Navigate to the folder "Generic directory" in the Identity Center tree and select the "Create LDAP organization" template.

5. Choose "Next >".

Select the repository "Tutorial-LDAP".
6. Choose "Next >".

![Pass wizard - Fill in constants](image)

Leaving the field "Organization name" empty as we will reference the repository constant directly.

The other fields are disabled to show that the values are retrieved from the repository definition.

7. Choose "Next >" and then "Finish".

8. Select the pass in the console tree and select the "Destination" tab:

![Identity Center - [Console Root] SAP NetWeaver Identity Management (Provisioning)](image)

The repository constants you specified when running the wizard are inserted in the fields. Replace the values for "dn" and "o" with the repository constants you added.

9. Choose "Apply".
Running the job

Run the job and verify that the organization is created in the directory server. Use an LDAP client to view the contents of the directory server.

If the job fails, inspect the log files, verify that you have specified the correct credentials for the directory server.

Adding a repository definition for the file system folder

Create a folder in the file system where you want to store the files that we create for each entry in the identity store. Here we create a folder files in C:\Program Files\SAP\IdM\Identity Center\Tutorial\Data source\.

The file system folder is added as a generic repository:

1. Select the Identity Center's "Repositories" entry and choose New/Repository from the context menu. Choose "Next >".

Navigate to the "Repositories" directory and select the "Generic repository" template.
2. Choose "Next >".

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

4. Add one repository constant:

   Name the constant *PATH* and enter the folder name as the value. You can use the global constant as part of the folder name.

   **Note:**
   *Make sure this folder exists.*

5. Choose "OK" to close the dialog box.
Section 3: Adding the create LDAP user tasks

In this section we will add and test the tasks for creating a user in the directory server. The top-level task will be called as the provisioning task from the LDAP privilege that we will create later. To easily identify the tasks that are called from the privilege, we use the following syntax:

`#{<Repository type>_<Operation>}

For instance:

`#LDAP_Provisioning
#LDAP_Modify
#FILE_Deprovisioning`

Creating a folder for the LDAP tasks

First, we create a folder that we will use for the tasks for the LDAP server.

1. Select the "Tutorial" identity store and choose New/Folder... from the context menu.

Enter LDAP as name for the folder.

2. Choose "OK". The folder is included in the console tree.

Deselect "Show folder in workflow" as the tasks in this folder should not be displayed in the workflow.

3. Choose "Apply".
Adding task: #LDAP_Provisioning

Next, we create the ordered task group "#LDAP_Provisioning".

1. Select the folder you just created and choose **New/Ordered task group** from the context menu.

   Modify the task name in the console tree.

   Fill in the fields with the following values:

   **Repository**
   Select "Tutorial-LDAP" in the list. Later on we will move the repository definition to the privilege, but to be able to test the tasks in an easy way, we specify the repository directly on the task.

2. Choose "Apply".
Adding task: Create LDAP entry

The next step is to create the action task "Create LDAP entry". This task is creating the user in the LDAP directory, although without any attributes, except the mandatory sn. The other attributes are added by the "#LDAP_Modify" task.

**Note:**
This sample shows usage of an iPlanet directory server. If you are using another directory server, you may need to use different values.

1. Select the task "#LDAP_Provisioning" and choose New/Action task/Run wizard to start the job wizard.
2. Choose "Next >".
3. Navigate to the "Generic directory" folder in the "Identity Center" tree and select the "Create LDAP InetOrgPerson" template.
4. Choose "Next >".
5. Select the "Tutorial-LDAP" repository.
Section 3: Adding the create LDAP user tasks

SAP NetWeaver Identity Management Identity Center Tutorial - Provisioning

4. Choose "Next >".

![Image of Job wizard - Fill in constants]

The values for most constants are retrieved from the corresponding repository constants. You only need to supply the name of the organization. Use the context menu to insert the repository constant referencing the organization name.

5. Choose "Next >" and then "Finish".

The task is included in the console tree:

![Image of console tree]

Rename this task to Create LDAP entry.

Although you selected a repository for this task, this is not included in the "Repository" field. This is because the parent task already contains a reference to this repository.

6. Choose "Apply".
7. Select the job in the console tree:

Modify the job name in the console tree.

Modify the job properties:

**Enabled**
Select this check box to enable the job to be run by a dispatcher.

**Run by dispatchers**
Select a dispatcher that should be responsible for running this job.

8. Choose "Apply".
9. Select the pass in the console tree and select the "Destination" tab:

Modify the pass name in the console tree.

Modify the definition created by the template.

```
  dn : %rep.DN%
  objectClass: top|person|organizationalPerson|inetOrgPerson
  sn: %MX_LASTNAME%
```

Use the context menu to insert the source attributes and the constants used in the definitions.

The prefix . (period) in front of objectClass ensures that this attribute will not be updated in a Modify operation.

Remove the redundant lines created by the template.

Add the line:

```
  changeType: Add
```

This line ensures that the job fails if trying to create an already existing entry.

10. Choose "Apply".
Testing the tasks

You are now ready to test the "#LDAP_Provisioning" task. First, we will just test the provisioning logic, and check that the dispatcher is able to process and expand the tasks. Then, we will test with an entry from the identity store.

Note: Make sure that the dispatcher is running.

Testing the task logic

1. Select the "#LDAP_Provisioning" task in the console tree and choose Test provisioning task… from the context menu:

When using an empty MSKEYVALUE, no jobs are run. Only the task structure is verified.

2. Choose "OK".

As the dispatcher processes the tasks, you see the log is being updated.

3. Choose "Close" to close the dialog box.
**Testing with an actual entry**

First, you need to find the MSKEYVALUE of an entry in the identity store that we can use for the test. We have already assigned the employee ID to the MSKEYVALUE and you can use any of these.

Test the task:

1. Select the "#LDAP_Provisioning" task in the console tree and choose "Test provisioning task…" from the context menu:

![Test provisioning task](image)

Enter the MSKEYVALUE of one of the entries in the identity store.

2. Choose "OK".

![Audit log](image)

View the log as the dispatcher processes the tasks.

3. Choose "Close" to close the dialog box.

*Note:*
*Repeat this for a few more entries, as we will need them later in the testing.*

View the contents of the directory server to verify that the entry has been created.
Troubleshooting

If any problem should occur during the execution, you can check some of the following:

- Verify that the dispatcher is running and that it is enabled for provisioning jobs.
- Verify that all tasks and jobs are enabled.
- Verify that the job has been defined for the given dispatcher.
- Verify that the directory server is available, and that the correct credentials are used.
- View the logs.
  - System log
    Verify that the dispatcher has requested the given job.
  - Job log
    View any error messages in the job log to see if you can find the cause of the problem.
- If you need to investigate a job more thoroughly, you can specify a different log file name for the job in the "Logging" tab of the job properties. You can also deselect the check box "Reset output file" to avoid overwriting the log file each time the job is run. This can be useful when debugging a provisioning job that may be run several times in sequence.
- If you need more logging info from a specific job, you can create a specific dispatcher and increase the log level in the dispatcher's .prop file. Specify that the job is to be run by this specific dispatcher. Make sure that the dispatcher is not running. To run the job, start the dispatcher from the command line with the following command:

  
  ```
  dispatcher_service_<dispatcher name> test runonce
  
  ```

  The job will then be run once and a detailed log file will be created.
Adding task: #LDAP_Modify

When the tasks execute without problems, we can continue by adding a task to modify the LDAP attributes.

Note:
When the repository is an LDAP server, it would not be necessary to use a different task for this, as the job is able to create or modify an entry, based on its existence. We use two tasks to make the example more generic.

Create the task in the same way as the "Create LDAP entry" task:

1. Select the "#LDAP_Provisioning" task (not the job) and choose New/Action task/Run wizard… from the context menu. Use the "Create LDAP InetOrgPerson" template. Use the repository constant as the organization name.

2. Modify the task name in the console tree.

Select "Public task" as we are going to call this task from the privilege we will create later on.

2. Choose "Apply".
3. Select the job in the console tree:

Modify the job name in the console tree.
Enable the job, and select the correct dispatcher.

4. Choose "Apply".

5. Select the pass in the console tree and select the "Destination" tab:

Modify the pass name in the console tree.
Remove the redundant lines and modify the remaining pass definitions as shown above.
Section 3: Adding the create LDAP user tasks

SAP NetWeaver Identity Management Identity Center Tutorial - Provisioning

The definition changeType=modify indicates that this pass will always perform a modify operation.

6. Choose "Apply".

Test the "#LDAP_Provisioning" task with an entry that is not present in the directory server. This will add more attributes to the entry, like displayName, givenName, title and location.

You can also test the "#LDAP_Modify" task separately. First, you must select the LDAP server as repository for the task. Enter an entry ID that was created before the "#LDAP_Modify" task was added. Remember to remove the repository again after you have tested the task.
Section 4: Adding the LDAP privilege

In this section we will add the privilege that contains links to the provision and deprovisioning tasks. We will also add the workflow task that is used to add the privilege to the entries in the identity store and see that the provision task is executed.

Creating the LDAP privilege

To add the privilege:

1. Select "Identity store metadata\Privileges" and choose New/Privilege… from the context menu.

Name
Enter "LDAP" as name of the privilege.

Repository
Select the repository "Tutorial-LDAP" as repository for this privilege. By adding the repository reference to the privilege, you could re-use the tasks for other privileges controlling other directory servers.
2. Select the "Tasks" tab:

Provisioning task
Choose "…" to the right of the field and locate the task "#LDAP_Provisioning".

Modify task
Choose "…" to the right of the field and locate the task "#LDAP_Modify".

If you only want the modify task to be executed for a subset of the attributes, you can select the attributes from the list.

3. Choose "OK".
Removing the repository from the top-level task

As we already included the repository reference in the top-level task ("#LDAP_Provisioning"), we have to remove this reference now that we have specified the repository reference in the privilege.

1. View the properties of the "#LDAP_Provisioning" task:

   Select "None" in the "Repository" field.

2. Choose "Apply".
Configuring the Workflow interface

Before we can use the Workflow interface we must do some initial configuration.

1. Select the "Tutorial" identity store in the console tree and select the "Workflow" tab:

Select "Identity store" as "Authentication method".

2. Choose "Apply".

3. Choose "Add user…".

Select "MX_PERSON" as "Entry type".

Fill in a user name and password you will use to log in to the Workflow interface.

4. Choose "OK".
Hiding the "Provisioning folder"

We will not use this folder, and we can hide it from the Workflow web interface.

1. Select the "Provisioning folder" in the console tree.

   ![Console tree with Provisioning folder selected](image)

   Deselect "Show folder in workflow".

2. Choose "Apply".

Creating the folder

We will create a separate folder for the workflow task:

1. Select the "Tutorial" identity store in the console tree and choose New/Folder… from the context menu.

   ![Folder properties dialog](image)

   Enter "Workflow" as name for the folder.

2. Choose "OK".
The folder is included in the console tree:

Select "Automatically expand folder" to specify that the tasks in this folder are automatically displayed when you log on to the workflow interface.

3. Choose "Apply".

Creating the workflow task

The next step is to create the workflow task that we will use to add the privilege to the entries in the identity store:

1. Select the "Workflow" folder and choose **New/Ordered task group** from the context menu.

Modify the task name in the console tree.
Select "Show on welcome page".

2. Select the "Attributes" tab:

Select "MX_PERSON" as entry type and configure the attributes for the task as displayed above.

3. Choose "Apply".

4. Select the "Access control" tab and choose "Add…".

Select "Logged-in user or identity store entry" in the "Allow access for" list. Enter the name of the identity store user you added previously.
5. Choose "OK".

The resulting access control is displayed in the details pane:

6. Choose "Apply".
Adding the privilege to the identity store entry

**Note:**
Make sure the Workflow web interface is configured for the Identity Center you are using.

We will now start the Workflow web interface and add the privilege to the identity store entry.

1. Start the Workflow web interface from the "Start" menu.
2. Choose "Login".

Fill in user name and password
3. Choose "Login".
4. Choose "Manage privileges" in the list of available tasks.

5. Choose "Search" to display all entries in the identity store.
6. Select one of the entries that are not added to the directory server.
7. Choose "…" to the right of the "Member of Privileges" field.

Choose "Search" and select the "LDAP" privilege that is displayed.
8. Choose "OK".

The privilege is added to the entry.
9. Choose "OK". You return to the task list. When the task completes successfully, the progress indicator turns green.

Verify that the entry is added to the directory server and that the jobs have completed successfully by inspecting the Identity Center's job log in the console tree.
Section 5: Adding the delete LDAP user tasks

In this section we add the tasks for deleting a user in the directory server. We will also update the privilege to contain a deprovisioning task link.

Adding task: #LDAP_Deprovisioning

First, we create the ordered task group "#LDAP_Deprovisioning":

1. Select the "LDAP" folder and choose New/Ordered task group from the context menu.

   Rename the task name in the console tree.

   Repository
   Select "Tutorial-LDAP" in the list to be able to test the task directly from the user interface.

2. Choose "Apply".
Adding task: Disable LDAP entry

This task is used to disable an entry. As there is no concept of disable in an LDAP directory, we change the displayName of the user, by adding the prefix "DISABLED". This is to visualize that the user has been disabled.

Note:
If you are using an LDAP client that does not show the displayName attribute, you may want to modify a different attribute.

An easy way of creating this task is to copy the "#/LDAP_Modify" task:
1. Select the "#/LDAP_Modify" task in the console tree and choose Copy from the context menu.
2. Select the "#/LDAP_Deprovisioning" task and choose Paste from the context menu.
3. Select the task in the console tree:
   - Rename the task to "Disable LDAP entry".
   - Deselect "Public task" as this task should not be available externally.
4. Choose "Apply".
Section 5: Adding the delete LDAP user tasks

5. Select the job in the console tree:

![Screenshot of the console tree showing job and pass selection](image)

- Rename the job to "Disable LDAP entry".
- Enable the job and select a dispatcher.

6. Select the pass in the console tree and select the "Destination" tab:

![Screenshot of the pass properties with "DISABLED:" added](image)

- Rename the pass to "Disable LDAP entry".
- Modify the pass properties to add the text "DISABLED:" in front of the display name.

7. Choose "Apply".
Adding task: Delete LDAP entry

The next task is "Delete LDAP entry":

1. Copy the "Disable LDAP entry" task.
2. Rename the task to "Delete LDAP entry".
3. Select the job:
   - Rename the job to "Delete LDAP entry".
   - Enable the job and select a dispatcher.
4. Choose "Apply".
5. Select the pass:
Rename pass to "Delete LDAP entry".
Modify the value of the changeType to "Delete". This specifies that the entry is deleted from the directory server.

6. Choose "Apply".

**Specifying a delay**

Additionally, we add a delay between disable and delete. Normally, this may be several days or even weeks, but in this case we specify one minute delay.

1. Select the "Delete LDAP entry" task.

Modify the task properties:

**Delay before start**

Enter / and select "Minutes" in the list.

2. Choose "Apply".
Testing the tasks

The tasks are now ready to be tested. Make sure that the dispatcher runs before you test the tasks.

1. Select the task "#LDAP_Deprovisioning" and choose Test provisioning task… from the context menu.

Use an empty MSKEYVALUE to test the task logic.

2. Choose "OK".

You can follow the task execution in the "Audit log" dialog box.

Note:
There will be a delay of one minute between "Disable LDAP entry" and "Delete LDAP entry".

3. Choose "Close".

Test the task with an entry ID that exists in the directory server. Observe that the display name of the entry is changed to, for instance, "DISABLED: Lisa Andersson" and then the entry is removed after a minute.
Updating the LDAP privilege

We can now add the deprovisioning task to the LDAP privilege:

1. View the properties of the LDAP privilege and select the "Tasks" tab:

   Select the "#LDAP_Deprovisioning" as "Deprovisioning task".

2. Choose "OK".

The provisioning and deprovisioning tasks of the privilege will be executed when the privilege is added or removed from a user. The provisioning task will not be executed for users which already have the privilege.
Removing the repository definition from the task

We can now remove the repository definition from the "#LDAP_Deprovisioning" task:

1. View the properties of the "#LDAP_Deprovisioning" task:

   Select "None" as repository for the task.

2. Choose "Apply".

Select "None" as repository for the task.

2. Choose "Apply".
Section 6: Resetting the tutorial data

During testing, the Identity Center database and the repositories are filled up with a lot of data. Especially the delta information may cause confusing results. Before we continue, we will reset the following:

- The directory server
- The identity store
- The delta information

As we add more repositories to the configuration, they need to be reset, too.

Resetting the directory server

We already have a job with a pass that creates the organization in the directory server. We are going to extend this job to reset all the data for the tutorial.

1. Copy the "Create LDAP organization" pass and ensure that the copy is placed above the original pass.
2. Select the new pass in the console tree and select the "Destination" pass:

Modify the pass name in the console tree.

Modify the pass definition.

3. Choose "Apply".

Note:

`changeType=deleteSubtree` is a very powerful function, which removes an entire subtree within an organization, with no warnings and no possibility to undo the deletion. Please use with care.
Emptying the identity store

There is a stored procedure in the Identity Center database that can be used to remove entries from the identity store. As a delete of a user would execute event tasks on the entry type and attributes, as well as saving the old values, the stored procedure deletes the entries without any event tasks being run. The entries are also removed from the historic data.

The stored procedure `mc_reset_ids_mskey` can be called from a To Database pass.

The syntax for the stored procedure is:
- Microsoft SQL Server: `execute mc_reset_ids_mskey <mskey>`
- Oracle: `call mc_reset_ids_mskey(<mskey>)`

`mc_reset_ids_mskey` has one parameter, the mskey to the entry that is to be removed. You can use the "Source" tab to define the selection and use `%MSKEY%` to call the procedure for each MSKEY that is found.

1. Select the "Create LDAP organization" pass and choose **New/To database** from the context menu. Select the "Source" tab.
   - Modify the pass name in the console tree to "Empty identity store".
   - Select "Use identity store" and select the "Tutorial" identity store in the list.

2. Choose "Build SQL query…".

   ![Build SQL query](image)

   Fill in the following values to specify that all entries with entry type MX_PERSON will be removed from the identity store, but not the entry with MSKEYVALUE Admin (the user that was given access to the workflow task).

   Select "MX_ENTRYTYPE" from the "Attribute name" list and "MX_PERSON" from the "Filter" list.

   Select "AND" from the "Operator" list, "MSKEYVALUE" in the "Attribute name" list, select the "Not" option button and enter "Admin" in the "Filter" field.

   Select "Include disabled entries" to make sure any disabled entries also are removed from the identity store.

3. Choose "OK".
The resulting query will look like this:

4. Select the "Destination" tab and fill in the following information:

Database
Use the context menu to insert the system parameter \%$ddm.identitycenter\%.

SQL updating
Select this check box to specify that the definitions are SQL statements.

Definitions
This definition calls the stored procedure mc_reset_ids_mskey with the correct parameters.

Note:
The syntax shown above is for Microsoft SQL Server. The syntax for Oracle is call mc_reset_ids_mskey(\%MSKEY\%)

5. Choose "Apply".
Section 6: Resetting the tutorial data

SAP NetWeaver Identity Management Identity Center Tutorial - Provisioning

Resetting the delta information

We have defined a delta database for the pass "HR to ID store". The delta database must be reset to avoid confusing results when testing the jobs and tasks. There is a stored procedure for removing all delta and audit trail information, \textit{mc\_reset\_delta\_nostatus}.

This stored procedure has one parameter, the delta identifier. You will find this name in the pass properties.

1. Select the "Empty identity store" pass and choose \textbf{New/To database} from the context menu:

   ![Image of Identity Center console](image.png)

   Rename the pass in the console tree.

   Select the "Destination" tab and fill in the pass properties:

   **Database**
   Use the context menu to insert the system parameter \%$ddm\_identitycenter\%.

   **SQL updating**
   Select this check box to specify that the definitions are SQL statements.

   **Definitions**
   This line calls the stored procedure \textit{mc\_reset\_delta\_nostatus} with the correct parameters.

   \textbf{Note:}
   \textit{The syntax shown above is for Microsoft SQL Server. The syntax for Oracle is call mc\_reset\_delta\_nostatus(TutorialHR)}

2. Choose "Apply".
Running the job

To run the job, view the job properties and choose "Run now". View the job log to verify that no errors occur during the execution of job.

The Identity Center's system log contains messages from the stored procedures confirming that the identity store and the delta information have been reset.

By enabling and disabling passes in the job, you can reset only selected parts.
In this section we will see how we can automatically assign the privilege when the users are added to the identity store instead of using the Workflow task to manually assign the privilege.

Updating the "HR to identity store" job

If we add the privilege to the entries when they are written to the identity store, they will automatically be created in the directory server when they are added to the identity store.

To do this, we must update the job that writes the entries to the identity store:

1. Select the "Destination" tab of the "HR to ID store" pass in the "HR to identity store" job:

Add the last line in the "Definitions" section:

**Attribute**
You can choose "Destination attributes" from the context menu to retrieve the attribute name MXREF_MX_PRIVILEGE that is the attribute containing the links to the privileges.

**Value**
Enter `<LDAP>` as the value. "LDAP" is the name (MSKEYVALUE) of the privilege we are using. By enclosing the value in brackets `< >`, you can reference the MSKEYVALUE directly. Otherwise you would have to reference the MSKEY.

2. Choose "Apply".

Running the job

If necessary, run the reset job to ensure that the identity store and directory server are empty.
Then run the "HR to identity store" job to create the entries in the identity store and verify that they are created in the directory server as well.
Section 8: Adding the file system tasks

This section shows how to add the tasks for updating the file system. For the example, this will be a folder in the file system, where files are created, written and deleted using the action tasks.

Adding task: #FILE_Provisioning

Create the task structure:

1. First, we create a folder for the tasks below the identity store node. Name the folder "File" and specify that it will not be displayed in the Workflow web interface.

2. Create an ordered group in this folder:

   Rename the task in the console tree to "#FILE_Provisioning".

   Select "Tutorial-FILE" as repository for the task to be able to test it before we add the privilege.

3. Choose "Apply".
Adding sub-task: Add file

This task is used to create the file for the given user. The EmployeeID will be used as file name.

1. Select the "#FILE_Provisioning" task and choose New/Action task/Empty job from the context menu.

   ![Image](image.png)

   Rename the task to "Add file" in the console tree.

   This task will inherit the repository from the parent task.

2. Select the job in the console tree:

   ![Image](image.png)

   Rename the job to "Add file" in the console tree.
Enable the job and select a dispatcher.

3. Choose "Apply".

4. Select the job and choose **New/Shell execute** from the context menu to add a pass to the job.

Select the "Destination" tab and enter the following definitions:

```
cmd /c echo User created %%ddm.date% %%ddm.time% > "%$rep.PATH\%MSKEYVALUE%.txt"
```

**Note:**
The file name must be enclosed by double quotes.

This command will add the text "User created <date> <time>" to a file with MSKEYVALUE of the given user as file name, for example 3001.txt for Lisa Andersson.

**Note:**
This will be different if running the job on a different platform than Microsoft Windows.

5. Choose "Apply".

**Testing the tasks**

Select the task "#FILE_Provisioning" and choose **Test provisioning task…** from the context menu and enter the entry ID of Lisa Andersson (3001).

A file called 3001.txt (or the MSKEYVALUE corresponding to the entry) in the folder you specified for the repository.

The contents of the file will be something like this:

```
User created 11.03.2003 15:13:51
```
Section 8: Adding the file system tasks

Adding task: #FILE_Modify

This task is very similar to the "Add file" task, so we can copy this task and modify it to create the "#FILE_Modify" task.

1. Make a copy of the "Add file" task.

Modify the task name in the console tree.
Select "Public task" as this task will be called as the Modify task from the privilege.

2. Choose "Apply".

3. Select the job:

Rename the job in the console tree.
Enable the job and select a dispatcher.

4. Choose "Apply".

5. Select the "Destination" tab of the pass:

   ![Image of console tree with file path and command entries]

   Rename the pass in the console tree.

   Enter the following definitions:

   ```cmd
   cmd /c echo User modified %ddm.date% %ddm.time% > "$rep.PATH\%MSKEYVALUE%.txt"
   cmd /c echo User name: %MX_FIRSTNAME% %MX_LASTNAME% >> "$rep.PATH\%MSKEYVALUE%.txt"
   cmd /c echo Email: %MX_MAIL_PRIMARY% >> "$rep.PATH\%MSKEYVALUE%.txt"
   ```

   We have still not added the e-mail address, so this information will be empty at the moment, but will be included when it is available.

6. Choose "Apply".

   This will replace the existing contents of the file.

   Redo the test from the previous section, and verify that the 3001.txt file now contains the following:

   ```
   User modified 11.03.2003 15:15:54
   User name: Lisa Andersson
   Email: 
   ```
Adding task: #FILE_Deprovisioning

The final step is to create the "#FILE_Deprovisioning" task below the "File" folder.

1. This task is similar to the "Add file" task. Make a copy of this task below the "#FILE_Provisioning" task.
2. Select the task:

Modify the task name in the console tree.
Select "Tutorial-FILE" as repository to be able to test the task before we add the privilege.
Select "Public task" as this task will be called as the deprovisioning task from the privilege.
3. Choose "Apply".
Section 8: Adding the file system tasks

SAP NetWeaver Identity Management Identity Center Tutorial - Provisioning

4. Select the job:

   ![Console tree image]

   Modify the job name in the console tree.

   Enable the job and select a dispatcher.

5. Choose "Apply".

6. Select the pass in the console tree:

   ![Console tree image]

   Modify the pass name in the console tree.

   Enter the following definitions:

   `cmd /c Del "%rep.PATH%\MSKEYVALUE%.txt"`
Note:
Be sure to verify that the parameters are correct, so that you do not accidentally delete the wrong file. If you are unsure, you should verify first, by using the following cmd statement in the pass:

```
cmd /c echo Deleted > "%$rep.PATH\%MSKEYVALUE%.txt"
```

that will write the text "Deleted" to the specified file.

7. Choose "Apply".

Testing the tasks

Test the "#FILE_Deprovisioning" task by choosing Test provisioning task… from the context menu. Use an empty MSKEYVALUE to verify that the logic is correct. Enter the MSKEYVALUE 3001 (the employee ID for Lisa Andersson). The file should be deleted from the folder.

Updating the "Reset tutorial data" job

We can add a pass to the "Reset tutorial data" job that removes the files from the directory.

1. Select the "Reset tutorial data" job and choose New/Shell execute from the context menu. Select the "Repository" tab:

Modify the pass name in the console tree.

Select "Tutorial-FILE" as repository for this pass.
2. Select the "Destination" tab:

Enter the following definition:

```
cmd /c Del "%rep.PATH%\*.txt"
```

that will remove all files in the directory. Use the repository constant to point to the correct directory.

**Note:**
Make sure that it is the path is correct in the same way as you did with the "Delete file" task. (See page 76.)

3. Choose "Apply".

4. Run the job and verify that all tutorial data is removed.
Section 9: Adding the FILE privilege

In this section we will add a privilege for the file system folder. As opposed to the LDAP privilege, we will add the FILE privilege by running a job that updates entries in the identity store based on the value of an attribute.

Creating the FILE privilege

To add the privilege:

1. Select "Identity store metadata\Privileges" and choose New/Privilege… from the context menu.

   ![Privilege window]

   **Name**
   Enter "FILE" as name of the privilege.

   **Repository**
   Select "Tutorial-FILE" as repository for this privilege.
2. Select the "Tasks" tab:

![Image](image)

**Provision task**
Choose "…" to the right of the field and locate the task 
"#FILE_Provisioning".

**Deprovisioning task**
Choose "…" to the right of the field and choose the "#FILE_Deprovisioning" task.

**Modify task**
Choose "…" to the right of the field and locate the task 
"#FILE_Modify".

3. Choose "OK".

**Removing the repository reference from the tasks**
We no longer need the direct repository reference on the top-level tasks, so we can remove them.

1. View the properties of the task 
"#FILE_Provisioning".
2. Select "None" as repository.
3. Choose "Apply".
4. Repeat the procedure for the task and 
"#FILE_Deprovisioning".
Assigning the FILE privilege to the entries

We will now create the job that assigns the privilege to entries in the identity store. We will assign the privilege based on the value of the MX_ADDRESS_CITY attribute.

1. Select the folder "Identity Center tutorial" and choose **New/Empty job** from the context menu.

2. Choose "Apply".

3. Select the job and choose **New/To identity store** from the context menu. Select the "Source" tab:
Rename the pass in the console tree.
Choose "Use identity store" and select "Tutorial" in the "Identity store" list.

4. Choose "Build SQL query":

```
Select "MX_ADDRESS_CITY" as the "Attribute name" and enter TRONDHEIM as the "Filter".
```

5. Choose "OK".

This results in the following query:

```
SELECT DISTINCT mskey FROM mxiv_sentries WHERE is_id=2 AND
((mskey IN (SELECT mskey FROM mxiv_sentries WHERE attrname='MX_ADDRESS_CITY' AND
searchvalue = 'TRONDHEIM'))) )
```

that will select all entries in the identity store with the value "TRONDHEIM" in the attribute "MX_ADDRESS_CITY".

6. Select the "Destination" tab:

Select "Tutorial" in the "Identity store" list.
Select "MX_PERSON as "Entry type".
Enter the definitions as shown above. This will add the "FILE" privilege to the entries selected by the query on the "Source" tab.

7. Choose "Apply".
Running the provisioning system

You should now be able to run the complete provisioning system:
1. If necessary, start by running the reset job.
2. Start the provisioning system by running the "HR to identity store" job.
3. When the job has completed, run the "Assign FILE privilege" job.
4. Verify that the job runs and the files are created as intended.
You can follow the progress, for instance in the Identity Center's job status overview or job log.

Testing modify

You can also test that the provisioning system works when updating an attribute.
1. Modify an entry in the hr.csv file. Edit the file with a text editor like Notepad.
2. Run the "HR to identity store" job.
3. Follow the progress in the Identity Center's job status and verify that you find the updated values in the directory server and generated files.
4. If you have changed the location of an entry, you must run the "Assign FILE privilege" job.
Section 10: Handling deleted entries

The next step is to handle entries that are deleted from the HR file. This is done by updating the "HR to identity store" job.

Disabling deleted entries

We will use the delta information to disable deleted entries in the identity store:

1. Select the "HR to ID store" pass and choose New/To identity store from the context menu and select the "Source" tab.

   ![Screenshot of the console tree with SQL query and interface elements]

   Rename the pass in the console tree.

   Use the context menu and select the system constant in the "Database" field.

   The SQL query selects all entries that are marked for delete in the delta table:

   ```sql
   SELECT Logentries.DN
   FROM Delta_Defs INNER JOIN Logentries ON Delta_Defs.OwnerID = Logentries.OwnerID
   WHERE (Delta_Defs.Name = 'TUTORIALHR') AND (Logentries.Operation = 4 OR Logentries.Operation=5)
   ```
2. Select the "Destination" tab:

Select "Tutorial" in the "Identity store" list.
Select "MX_PERSON" as "Entry type".

The definitions update each of the entries matching the query on the "Source" tab.
The source attribute for MSKEYVALUE is here %DN%, as the source is the delta database.
The attribute MXREF_MX_PRIVILEGE is a multi-value attribute that is used to hold the privileges. When the entry is disabled, we also want to remove the privileges. This is done by the following definition:

\[
\text{MXREF_MX_PRIVILEGE } \{R\}
\]

The \{R\} removes all values from a multi-value attribute.

When the privilege(s) is removed, the deprovisioning task on the privilege(s) are executed.
The entries are disabled by setting the attribute MX_DISABLED to true (1).

Note:
This definition must be the last one, because otherwise the entry is disabled before the privilege attribute is updated.

2. Choose "Apply".
Updating the "Reset tutorial data" job

The "Empty identity store" pass of the "Reset tutorial data" job does not handle disabled entries. We need to do the following to include disabled entries:

1. Select the "Source" tab of the "Empty identity store" pass.
2. Choose "Build SQL query…".
3. Choose "OK" to close the dialog box.
4. Choose "Apply" to update the pass.

Select "Include disabled entries".
Testing the mechanism

If necessary, run the "HR to identity store" and "Assign FILE privilege" jobs at least once to make sure that the users are provisioned to the LDAP server and that the files are created in the file system.

1. Open the file *hr.csv* that we read with the "HR to identity store" job and remove one of the entries (you may want to take a copy of the file first, to keep the complete file under a different name).

   If you remove one of the entries with location Trondheim, you will see that the file created in the file system is removed too.

2. Run the "HR to identity store" job again.

3. View the job log to see that the deprovisioning jobs are run after the identity store is updated:

   ![Job Log Screenshot]

4. Verify that the entry is removed from the LDAP server and that the file is removed from the file directory.
Section 11: Adding the telephone numbers

The phone numbers and e-mail addresses are added in the same way as the HR data. First the data will be read (using a From ASCII pass) from the tel.csv file into a temporary database table (Tutorial_TEL). Another pass will update the entries in the identity store.

Adding a repository definition for the tel.csv file

Add the tel.csv as a repository definition in the same way as the hr.csv file. (See page 5.) Name the repository "Tutorial-TEL".
Adding the telephone numbers

The "TEL to identity store" job consists of three passes. The first pass reads the tel.csv file. The second pass updates the identity store and the third pass removes any deleted telephone numbers.

1. Select the folder "Identity Center tutorial" and choose **New/Empty job** from the context menu.

   ![Image of Identity Center tutorial](image)

   Modify the job name in the console tree.

   Enable the job.

   Select a dispatcher for the job.

2. Choose "Apply".
3. Select the job in the console tree and choose **New/From ASCII file** from the context menu:

Modify the pass name in the console tree.
Select the repository "Tutorial-TEL".

4. Select the "Source" tab:

Use the context menu to insert the repository constant for the file name.
Enter , (comma) as the field separator.
Select "Header line" to specify that the first line in the file is a header line that should not be written to the destination.
5. Select the "Destination" tab:

Use the context menu to insert the system parameter \%Sddm.identitycenter\% as the database name.

Enter TUTORIAL_TEL as the table name.

Use the data source template to insert the pass definition.

6. Choose "Apply".

**Running the job**

Make sure the dispatcher is running. View the job properties and choose "Run now" to run the job. The table "TUTORIAL_TEL" is created in the Identity Center database and will contain the telephone numbers from the tel.csv file.
Section 11: Adding the telephone numbers

SAP NetWeaver Identity Management Identity Center Tutorial - Provisioning

Updating the identity store

The next pass in the job will add the telephone numbers to the entries in the identity store.

1. Select the "Read telephone numbers" pass and choose New/To identity store. The pass is included in the console tree. Select the "Source" tab:

Modify the pass name in the console tree.

Use the context menu to insert the system parameter in the "Database" field.

Fill in the SQL statement to return all rows from the table created by the "Read telephone numbers" pass.

2. Select the "Destination" tab:
Fill in the information about the identity store.

Choose "Insert template" and select "Data source template" to insert the definitions for the pass.

Add the statement:

`changeType: modify`

This is done to ensure that no entries are added to the identity store based on the data in the `tel.csv` file. It is the `hr.csv` file that is the master data source. This job should only add attributes to existing entries.

Use the context menu to insert the destination attributes from the entry type that corresponds to the attributes you add to the entries.

3. Choose "Apply".

**Note:**
If the `tel.csv` contains entries (identified by the EmployeeId) which are not in the identity store, the job will fail. This error message can be safely ignored.

### Updating the "#LDAP_Modify" task

The new attributes must be added to the "#LDAP_Modify" task to include them in the directory server.

1. Select the "Destination" tab of the "Modify LDAP entry" pass in the "Modify LDAP entry" job:

![Diagram of Identity Center with highlighted "Create LDAP entry" step and "Modify LDAP entry" step]

Add the following attributes to the definition:

- `telephoneNumber %MX_PHONE_PRIMARY%
- `facsimileTelephoneNumber %MX_FAX_PRIMARY%
- `mail %MX_MAIL_PRIMARY%

2. Choose "Apply".
Running the job

Make sure the dispatcher is running. View the properties of the "TEL to identity store" job and choose "Run now" to run the job. When the entries in the identity store are updated, the "Modify tasks" we defined on the privileges are executed, and the e-mail addresses and telephone numbers are added to the entries in the LDAP server and to the files.

Enabling the delta

We can now enable the delta database. We add the delta database to the "Add telephone numbers" pass. Then, the delta database is used as a source for a To-pass that will delete the mail attributes if they are removed from the source.

1. Select the "Add telephone numbers" pass in the "TEL to identity store" job. Select the "Delta" tab:

   ![Image of the Delta tab]

   Fill in the fields with the following values:

   **Enable delta**
   Select this check box to enable the delta on this pass.

   **Delta database**
   Use the context menu to insert the system parameter %$ddm.identitycenter% to specify that you want to store the delta database in the Identity Center database.

   **Delta identifier**
   Enter TutorialTEL as the delta identifier.

   Do not enable automatic deletion, as the tel.csv file is not the master data source for the identity store. No entries should be deleted.

2. Choose "Apply".
Run the job twice to verify that the job detects that there are no changes in the data source:
Removing deleted attributes

Currently, there is no automatic deletion of telephone numbers and attributes if they are removed from the `tel.csv` file. To do this, it is necessary to read from the delta database, to determine which entries were removed.

This pass reads the delta database created in the previous section and removes the attributes from the entries in the identity store if the attributes are removed from the data source.

1. Select the "Add telephone numbers" pass and choose **New/To identity store** from the context menu. Select the "Source" tab:

![Image of Identity Center](image)

Modify the pass name in the console tree.

Enter the following SQL statement that will select all entries from the delta database that have been deleted from the source in current or earlier runs:

```sql
SELECT Logentries.DN, Delta_Defs.OwnerID, Delta_Defs.Name, Logentries.Operation
FROM Logentries INNER JOIN
Delta_Defs ON Logentries.OwnerID = Delta_Defs.OwnerID
WHERE (Logentries.Operation > 3) AND (Delta_Defs.Name = 'TutorialTEL')
```

It is necessary to join the Logentries table and the OwnerID table to ensure that only entries with the correct Delta identifier are returned.

**Note:**

The name of the delta identifier is case sensitive.
2. Select the "Destination" tab:

Enter the definitions for the pass as shown above. The two percent signs are necessary to specify that an empty value is written to the identity store.

The statement

```
changeType modify
```

specifies that this pass should only modify existing entries, not add entries to the identity store.

3. Choose "Apply".

To test this pass, we must remove an entry from the tel.csv file.
Run the job. The attributes will be removed from the identity store and the "Modify tasks" defined on the privileges will remove the attributes from the entries in the LDAP server and the files in the file system."
Updating the "Reset tutorial data" job

The "Reset tutorial data" job needs to be updated to include removal of the delta database created by the "Read telephone numbers" pass.

1. Select the "Destination" tab of the "Empty delta database" pass in the "Reset tutorial data" job:

Add the following line to the definitions:

```
Execute mc_reset_delta_nostatus 'TutorialTEL'
```

This will remove the data from the delta database.

2. Choose "Apply.

Run the job to remove all data from the identity store, the directory server, the file system and the delta databases.

Running the provisioning system

Run the "HR to identity store" and "Assign FILE privilege" jobs to load all the initial data.

You should now verify that everything works as expected:

1. Update the hr.csv file by adding a new entry containing your own name at the end of the file. You have to define a new EmployeeID, for example 3051.

2. Select the "HR to identity store" job and choose "Run now". The following should happen:
   - The job runs and updates the identity store with the new entry.
   - The entry is added to the directory server.

3. If you added Trondheim as location, you can run the "Assign FILE" privilege job and verify that a file is generated for the entry.

4. Update the tel.csv file, by adding a new entry (using the same employeeID as above).
5. Select the job "TEL to identity store" and choose "Run now". The following should happen:
   - The job runs and updates the identity store with any changes from the tel.csv file.
   - The event handling defined on the privilege will execute the task "Modify user", which will cause the new attributes to be updated in the LDAP directory.

6. Verify that the LDAP directory contains the new information.

   Note:
   The LDAP client may be limited to show only the first 50 entries. The new entry may not be included in the result.

What's next?

Although the system now handles changes in the identity store, you could improve the system in different ways:

- You could add an event agent for each of the file repositories that automatically runs the jobs "HR to identity store" and "TEL to identity store". Then you would not need to start these jobs manually.
- Or you could schedule the jobs to run with specific intervals to make sure that any changes to the files are handled within a defined time frame.