SAP Identity Management Tutorial: Context-Based Assignments
Content

1 SAP Identity Management Tutorial: Context-Based Assignments .................................................. 4

2 About Context-Based Assignments ................................................................. 5

2.1 Context-Based Assignment Process .................................................. 6

2.2 Conditional and Default Contexts .................................................. 6

2.3 Use Cases ................................................................. 7

   Use Case 1: Basic Context-Based Provisioning ........................................ 8

   Use Case 2: Provisioning Using Default Context ................................... 9

   Use Case 3: Provisioning Using Conditional Context ............................ 10

2.4 Forms, Roles and Privileges ................................................................. 11

   Basic Context-Based Provisioning and Provisioning Using Default Context 12

   Provisioning Using Conditional Context ............................................ 13

2.5 The Data Sources ................................................................. 13

3 Configuration Steps ................................................................. 16

3.1 Creating a New Package ................................................................. 17

3.2 Defining the Package Constant TUTORIAL_TARGET .................................. 18

3.3 Creating Repository Type for the Target Folder .................................. 19

3.4 Creating Repository for the Target Folder .......................................... 21

3.5 Specifying the System Log Level .................................................. 22

3.6 Disabling Automatic Attribute Creation .......................................... 23

3.7 Creating Package JScript CreateFileName ........................................ 24

4 Building the Identity Store ................................................................. 27

4.1 Displaying the Details for Entries .................................................. 27

4.2 Reading the HR Data into the Identity Store ........................................ 28

   Creating the Job Folder and Adding the Job ...................................... 28

   Creating the Pass to Update the Identity Store .................................. 29

   Running the Job ........................................................................ 31

   Verifying the Content of the Identity Store ........................................ 32

4.3 Creating a Context Type ................................................................. 33

4.4 Adding Contexts, Roles and Privileges .......................................... 36

   Reading the ASCII File ................................................................ 37

   Running the Job ........................................................................ 39

   Updating the Identity Store ................................................................ 40

   Running the Job and Verifying the Contents of Identity Store ........... 42

   Viewing the Properties of the Tutorial Data ...................................... 44

4.5 Creating Assignment Request Forms ............................................. 49
1 SAP Identity Management Tutorial: Context-Based Assignments

This tutorial helps you understand the provisioning in SAP Identity Management 8.0 using context-based assignments.

Prerequisites

The following software is required:

- SAP Identity Management 8.0 SP00 or higher is correctly installed and licensed.
- SAP Identity Management User Interface is installed and configured, as described in Installing and Configuring SAP Identity Management User Interface. It also implies the administrator user with access to at least Self Services, Monitoring and Manage tabs in the User Interface.
- At least one dispatcher is configured and is running, as described in Configuring the Initial Dispatcher.
- The data source file used in this tutorial (ContextBasedAssignments_tutorial.csv) is downloaded. The file is available together with this document on the SAP Identity Management Community topic page. Navigate to: Implement > Documentation > Release 8.0

The Manual

This tutorial is not a substitution for training. Person names used in this tutorial are fictional.
About Context-Based Assignments

Context-based assignments involve three entries – a person, a role or a privilege, and a context. The purpose of context-based assignments is to reduce the number of roles.

To any role or privilege assignment it is possible to add a reference to a given context that limits the validity of the assignment to that specific context. A context may be a region, a project or an organizational unit. The typical scenario is a project or a store management.

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

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

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

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

Related Information

- Context-Based Assignment Process [page 6]
- Conditional and Default Contexts [page 6]
2.1 Context-Based Assignment Process

The backend system is updated using member event processes, typically defined as part of the repository definition. An Add member event process, which for example creates an account in the backend system, receives references to the user, the role and the context. It is responsible for assigning the correct authorizations in the backend system, based on the received information.

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

The legal context types must also be defined for the roles that use context. The legal context types are used for two purposes:

- The Identity Management User Interface will use this information when performing a direct assignment. A search for context will only be done in the defined entry types. When searching for a role or privilege, only the roles or privileges that are relevant for this context type are displayed.
- During inheritance, only roles or privileges that support the same context type as specified in the assignment are inherited. If the context is not allowed, the assignment is silently ignored.

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

2.2 Conditional and Default Contexts

Conditional Contexts

For privileges, it is possible to define a list of conditional contexts on the Context tab of the privilege properties (held by the MX_CTX_CONDITIONAL attribute). In this case, the privilege will only be assigned if one of the specified contexts also is part of the assignment. For example, when assigning a role with a region as context, only those privileges matching the given region will be assigned.

Default Contexts

It is possible to define default context(s) for each user by adding the MX_CTX_AUTO_VALUES attribute to the person entries. This attribute contains the default contexts for each context type. For example, this can be used to
define a default region for each user. This value is then used when assigning a role or privilege that uses region as context type. Only default contexts for the corresponding context type in question will be used.

The context configuration for a privilege or repository definition includes the policy for how this value is to be used. The check is performed for both direct and inherited contexts, and is only done for privileges (that is, not for roles).

2.3 Use Cases

Context-based provisioning to a non-context system (a system that is not context aware), is used for the use cases. This means that the Add member event processes, which are provisioning the users, need to “translate” the contexts to the system.

The following three use cases are defined:

- **Use Case 1: Basic Context-Based Provisioning** [page 8]
- **Use Case 2: Provisioning Using Default Context** [page 9]
- **Use Case 3: Provisioning Using Conditional Context** [page 10]

All use cases are based on a scenario with employees in a factory which are given rights (roles and privileges) based on their job role. In this tutorial, there is one system for all the factories, represented by a folder `FactorySystem` and the repository type `FACTORY_SYSTEM` in the Identity Management Developer Studio. The concept of context, when assigning privileges to a user, is introduced to this system by creating files with the following naming convention:

<cleaned MSKEYVALUE of the context>-<cleaned MSKEYVALUE of the privilege>-<MSKEYVALUE of the provisioned user>.txt. For example, FACTORY_Germany-PRIV_HRAdmin-3001.txt or COUNTRY_France-PRIV_ProductionPlantFR-3003.txt.

See more about the cleaned MSKEYVALUE in the Creating package JScript CreateFileName section. When a user is given a particular role (and thus one or more privileges) and a context, a file is created (containing the timestamp of when the privilege was assigned to the user) and provisioned to the `FactorySystem` folder. The files are representing the authorizations in the factory system.

**Note**

The system in this tutorial is represented by a `FactorySystem` folder for simplicity’s sake. It is possible to provision to other systems by replacing the To ASCII file passes used in this tutorial to passes writing to other systems.

The context and the role(s) for a given person are selected by using an assignment request form created in the Identity Management Developer Studio.

Related Information

Creating Package JScript CreateFileName [page 24]
2.3.1 Use Case 1: Basic Context-Based Provisioning

You use basic context-based provisioning to assign a role with a context to the user to be provisioned.

For basic context-based provisioning, you need to select both the context and the role(s) for a given person. You can do this by using the assignment request form **Context-Based Assignment Use Case 1 and 2**. For details about creating this form, see **Creating Assignment Request Forms**.

The context type used in this use case is FACTORY. The following context entries are defined:

- FACTORY: Norway
- FACTORY: Germany
- FACTORY: France

The roles that can be assigned to a person are the following:

- ROLE: HR_Manager
- ROLE: SupplyManager
- ROLE: Manager

For details about forms, roles and privileges used in this use case, see section **Forms, Roles and Privileges**.

Related Information

**Forms, Roles and Privileges** [page 11]
**Creating Assignment Request Forms** [page 49]
2.3.2 Use Case 2: Provisioning Using Default Context

Using default context is a variant of the basic context-based provisioning where the user to be provisioned has a defined default context (MX_CTX_AUTO_VALUES attribute on the person entry).

For provisioning using default context, you use the assignment request form Context-Based Assignment Use Case 1 and 2. For details about creating this form, see Creating Assignment Request Forms.

The context, roles and privileges for this use case are the same as for the basic context-based provisioning use case, and are described in the Forms, Roles and Privileges section.

Depending on the selected autoassign context policy on the repository definition FACTORY_SYSTEM (on the Context tab of the repository type), the default context will be added to the user when it is provisioned. The following autoassign context policies are available:

- **None**: No autoassign context policy is activated. That is, if this policy is selected, then the default context will not be added to the user when it is provisioned even if it is defined for the user.
- **If missing**: The defined default context will be automatically assigned to the user when provisioned, even when the context is not selected during the assignment (is missing). But the default context will only be added if a context is not selected during the assignment. If any context is selected during the assignment, then this is the context that is added to the user when it is provisioned and the assignment of the default context is not triggered.
- **Always**: The defined default context will always be assigned to the user when provisioned, given that the same context is not already assigned to the user. In other words, if a context is selected for the user during the assignment, then both this context and the default context will be added to the user when provisioned. This also means that the default context is added to the provisioned user even when the context is not selected for the user during the assignment, resulting in two authorizations in the system – one with and one without the context. The exception when the default context is not added to the user that is provisioned is when the context selected during the assignment is the same as the one defined as the default context.
2.3.3 Use Case 3: Provisioning Using Conditional Context

You use the conditional context to assign only those privileges for which a given context is present.

The attribute MX_CTX_CONDITIONAL is set for each privilege defined for a role. When the user at some point is assigned the role with a given context, then only the privilege that has a matching context will be assigned to the user through the role association (even if this role also has other privileges).

In this use case, the context type COUNTRY is used, and the following context entries are defined:

- COUNTRY:Norway
- COUNTRY:Germany
- COUNTRY:France

For details about forms, roles and privileges used in this use case, see section Forms, Roles and Privileges.

The assignment request form is different from the form used in the previous use cases (as assignment request forms are defined per entry type). You need to select the context and the role(s) for a given person by using the assignment request form Context-Based Assignment Use Case 3. Only one role can be assigned to the users in this use case: ROLE:ProductionSupervisor.

For details about creating this form, see Creating Assignment Request Forms.
2.4 Forms, Roles and Privileges

The following forms are defined to assign roles (and thus privileges) to the users:

Table 1:

<table>
<thead>
<tr>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context-Based Assignment Use Case 1 and 2</td>
<td>This assignment request form is used to assign roles (and privileges) with a context to the users in the identity store. The form is the same for use cases 1 and 2 (that is, basic context-based provisioning and provisioning using default context).</td>
</tr>
<tr>
<td>Context-Based Assignment Use Case 3</td>
<td>This assignment request form is used to assign roles (and privileges) with a context to the users in the identity store for use case 3 (conditional context).</td>
</tr>
</tbody>
</table>

Common for the three use cases are the two provisioning processes that are created, one for provisioning and one for deprovisioning of users for the repository definition FACTORY_SYSTEM.

Every time a user is given a particular privilege, a file will be created (containing the timestamp of when the privilege was assigned to the user) and provisioned to the respective folder:

Table 2:

<table>
<thead>
<tr>
<th>Process</th>
<th>Description</th>
</tr>
</thead>
</table>
| Factory_AddEntry             | This process is referenced from the FACTORY_SYSTEM repository type. It is defined on the Event Process tab as the Add member event process (Add field) of the repository type.  
                              | The process contains an action task Add File to FactorySystem Folder which creates a file containing the timestamp of when a privilege is assigned to user and provisions it to the FactorySystem folder. |
| Factory_RemoveEntry         | This process is referenced from the FACTORY_SYSTEM repository type. It is defined on the Event Process tab as the Remove member event process (Remove field) of the repository type.  
                              | The process contains an action task Delete File From FactorySystem Folder which deletes the previously created file from the FactorySystem folder. |

Use cases for basic context-based provisioning and for provisioning using default context are based on the same roles, privileges and the role/privilege hierarchy.

Use case 3 (provisioning using conditional context) is different.
2.4.1 Basic Context-Based Provisioning and Provisioning Using Default Context

There are three roles and three privileges defined for the two use cases: Basic context-based provisioning and Provisioning using default context.

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROLE:HR_Manager</td>
<td>This role gives the privilege PRIV:HRAdmin.</td>
</tr>
<tr>
<td>ROLE:SupplyManager</td>
<td>This role gives the privileges PRIV:Purchasing and PRIV:Distribution.</td>
</tr>
<tr>
<td>ROLE:Manager</td>
<td>This role inherits the privileges PRIV:Purchasing and PRIV:Distribution from its child role ROLE:SupplyManager and the privilege PRIV:HRAdmin from its child role ROLE:HR_Manager.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIV:HRAdmin</td>
<td>The privilege gives the user access to the necessary information and the human resource management systems.</td>
</tr>
<tr>
<td>PRIV:Purchasing</td>
<td>The privilege gives the user the authorization to perform purchasing of the materials and products needed for the production. It also gives access to necessary systems for purchasing.</td>
</tr>
</tbody>
</table>
2.4.2 Provisioning Using Conditional Context

There is one role and three privileges defined for the use case: Provisioning using conditional context.

<table>
<thead>
<tr>
<th>Role</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROLE:ProductionSupervisor</td>
<td>This role gives the privilege PRIV:ProductionPlantNO, PRIV:ProductionPlantDE and/or PRIV:ProductionPlantFR.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIV:ProductionPlantNO</td>
<td>This privilege gives the users the right to access and use the factory production plant in Norway.</td>
</tr>
<tr>
<td>PRIV:ProductionPlantDE</td>
<td>This privilege gives the users the right to access and use the factory production plant in Germany.</td>
</tr>
<tr>
<td>PRIV:ProductionPlantFR</td>
<td>This privilege gives the users the right to access and use the factory production plant in France.</td>
</tr>
</tbody>
</table>

2.5 The Data Sources

There are two data sources used in this tutorial:
A sample table HR_Sample in the Identity Management database.

A CSV file ContextBasedAssignments_tutorial.csv.

The HR_Sample table contains the basic information about the person objects (people in the organization). It contains the following attributes:

- EmployeeID
- LastName
- FirstName
- Title
- Dep (department)
- Location
- Tel
- Fax
- Email

<table>
<thead>
<tr>
<th>EmployeeID</th>
<th>LastName</th>
<th>FirstName</th>
<th>Title</th>
<th>Dep</th>
<th>Location</th>
<th>Tel</th>
<th>Fax</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>Andersson</td>
<td>Lisa</td>
<td>Secretary</td>
<td>Admin</td>
<td>Oslo</td>
<td>(00) 666 1904 (00) 666-1101 andersson.lisa@...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td>Bedson</td>
<td>Brad</td>
<td>CEO</td>
<td>Admin</td>
<td>Trondheim</td>
<td>(00) 666 1905 (00) 666-1101 bedson.brad@...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3003</td>
<td>Cabe</td>
<td>Bradley</td>
<td>Admin</td>
<td>Oslo</td>
<td>(00) 666 1906 (00) 666-1101 bradley.cabe@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4004</td>
<td>Hennson</td>
<td>Carter</td>
<td>Senior Consultant</td>
<td>Mandal</td>
<td>(00) 666 1967 (00) 666-1101 carter.hennson@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5005</td>
<td>Clarke</td>
<td>Allan</td>
<td>President</td>
<td>Consultancy</td>
<td>Mandal</td>
<td>(00) 666 1968 (00) 666-1101 allan.clarke@...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6006</td>
<td>Taylor</td>
<td>Mike</td>
<td>Boardroom Office</td>
<td>Oslo</td>
<td>(00) 666 1999 (00) 666-1101 mike.taylor@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7007</td>
<td>McKeen</td>
<td>Betty</td>
<td>Market Secretary</td>
<td>Marketing</td>
<td>Oslo</td>
<td>(00) 667 1900 (00) 666-1102 betty.mckeen@...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8008</td>
<td>Kelly</td>
<td>David</td>
<td>Publications Co.</td>
<td>Marketing</td>
<td>Oslo</td>
<td>(00) 667 1997 (00) 666-1102 david.kelly@...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9009</td>
<td>Woodgate</td>
<td>Audrey</td>
<td>Secretary</td>
<td>Oslo</td>
<td>(00) 667 1998 (00) 666-1102 audrey.woodgate@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10010</td>
<td>Rogers</td>
<td>Betsey</td>
<td>Secretary</td>
<td>Oslo</td>
<td>(00) 667 1999 (00) 666-1102 betsey.rogers@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11011</td>
<td>Robinson</td>
<td>Blake</td>
<td>Senior Consultant</td>
<td>Consultancy</td>
<td>Oslo</td>
<td>(00) 667 1941 (00) 666-1102 blake.robinson@...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12012</td>
<td>Batty</td>
<td>Philip</td>
<td></td>
<td>Oslo</td>
<td>(00) 667 1941 (00) 666-1102 philip.batty@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13013</td>
<td>Burns</td>
<td>Elizabeth</td>
<td>Accountant</td>
<td>Oslo</td>
<td>(00) 669 1942 (00) 666-1103 elizabeth.burns@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14014</td>
<td>Smith</td>
<td>Pamela</td>
<td>Chief Financial Officer</td>
<td>Finance</td>
<td>Oslo</td>
<td>(00) 667 1943 (00) 666-1104 pamela.smith@...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15015</td>
<td>Vinkcook</td>
<td>George</td>
<td></td>
<td>Mandal</td>
<td>(00) 667 1944 (00) 666-1104 george.vinkcook@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16016</td>
<td>Mills</td>
<td>Michael</td>
<td>System Engineer</td>
<td>Mandal</td>
<td>(00) 667 1945 (00) 666-1104 michael.mills@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17017</td>
<td>Prover</td>
<td>Jacob</td>
<td></td>
<td>Mandal</td>
<td>(00) 667 1946 (00) 666-1104 jacob.prover@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18018</td>
<td>Owens</td>
<td>Matthew</td>
<td></td>
<td>Mandal</td>
<td>(00) 667 1947 (00) 666-1104 matthew.owens@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19019</td>
<td>Raciu</td>
<td>Joseph</td>
<td>Program Engineer</td>
<td>Development</td>
<td>Mandal</td>
<td>(00) 667 1948 (00) 666-1104 joseph.racie@...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20020</td>
<td>Wolfe</td>
<td>Christopher</td>
<td></td>
<td>Mandal</td>
<td>(00) 667 1949 (00) 666-1104 christopher.wolfe@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21021</td>
<td>Murphy</td>
<td>Nicholas</td>
<td>Senior System Engineer</td>
<td>Development</td>
<td>Mandal</td>
<td>(00) 667 1950 (00) 666-1104 nicholas.murphy@...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22022</td>
<td>Thompson</td>
<td>Andrew</td>
<td></td>
<td>Mandal</td>
<td>(00) 668 1900 (00) 666-1102 andrew.thompson@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23023</td>
<td>Maxwell</td>
<td>William</td>
<td></td>
<td>Mandal</td>
<td>(00) 669 1901 (00) 666-1102 william.maxwell@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24024</td>
<td>Cole</td>
<td>Joshua</td>
<td>Senior Consultant</td>
<td>Consultancy</td>
<td>Bergen</td>
<td>(00) 669 2091 (00) 666-1102 joshua.cole@...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25025</td>
<td>Brown</td>
<td>David</td>
<td></td>
<td>Bergen</td>
<td>(00) 668 2092 (00) 666-1102 david.brown@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26026</td>
<td>Green</td>
<td>Hannah</td>
<td>Market Coordinator</td>
<td>Marketing</td>
<td>Bergen</td>
<td>(00) 668 2093 (00) 666-1102 hannah.green@...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27027</td>
<td>Chong</td>
<td>Emily</td>
<td></td>
<td>Bergen</td>
<td>(00) 668 2094 (00) 666-1102 emily.chong@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28028</td>
<td>Brindley</td>
<td>Madison</td>
<td>Sales Manager</td>
<td>Sales</td>
<td>Bergen</td>
<td>(00) 669 2095 (00) 666-1102 madison.brindley@...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29029</td>
<td>Oliver</td>
<td>Elizabeth</td>
<td></td>
<td>Trondheim</td>
<td>(00) 669 2096 (00) 666-1102 elizabeth.hol@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30030</td>
<td>Rogers</td>
<td>Alexas</td>
<td>Senior Program Manager</td>
<td>Development</td>
<td>Trondheim</td>
<td>(00) 669 2097 (00) 666-1102 alexas.rogers@...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31031</td>
<td>Davies</td>
<td>Sarah</td>
<td></td>
<td>Trondheim</td>
<td>(00) 669 2098 (00) 666-1102 sarah.davies@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32032</td>
<td>Porter</td>
<td>Taylor</td>
<td></td>
<td>Trondheim</td>
<td>(00) 669 2099 (00) 666-1102 taylor.porter@...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33033</td>
<td>Green</td>
<td>Leonas</td>
<td>Senior System Engineer</td>
<td>Development</td>
<td>Trondheim</td>
<td>(00) 669 2100 (00) 666-1102 leonas.green@...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The CSV file ContextBasedAssignments_tutorial.csv contains the contexts, privileges and roles needed in this tutorial. It also builds a role (and privilege) hierarchy, sets the contexts allowed for each role and privilege.
sets the conditional contexts on privileges for use case 3 (provisioning using conditional context) and defines the repository definitions on the privileges:

The CSV file does not define the actual context entry types FACTORY and COUNTRY, and the default contexts for users. These have to be defined manually. How to do this is described in the document.
3 Configuration Steps

Before you proceed with the tutorial, you need to perform the following configuration steps:

1. Create a new package that will contain all the configurations needed for this tutorial.
2. To be able to reference the files created in this tutorial in a uniform way, create a package constant containing the path to the directory where the target repository for the files (FactorySystem folder) is to be placed.
3. Create the FactorySystem folder, which will be a target repository for the files. For this you will create a new repository type in the Identity Management Developer Studio and a repository definition for the folder in the Identity Management Administration User Interface.
4. To be able to view the log information shown in this tutorial, you need to set the log level for the system log to Info.
5. Make sure to disable the automatic attribute creation option for the identity store.
6. Create the package Java script CreateFileName.
   When a user is given a particular role (and thus one or more privileges) and a context, a file is created (containing the timestamp of when the privilege was assigned to the user) and provisioned to the FactorySystem folder. The name of the file has the following naming convention:
   <cleaned MSKEYVALUE of the context>-<cleaned MSKEYVALUE of the privilege>-<MSKEYVALUE of the provisioned user>.txt
   For example, FACTORY_Germany-PRIV_Warehouse-3001.txt
   Cleaned MSKEYVALUE of the role or context is MSKEYVALUE where the colon ("\:" ) is replaced by the underscore ("\:"") – for MSKEYVALUE "PRIV:Warehouse" the cleaned MSKEYVALUE will be "PRIV_Warehouse", and for "FACTORY:Germany" it will be "FACTORY_Germany". The reason is that it is not possible to use the colon ("\:" ) in a file name. The Java script CreateFileName is used for this purpose – to construct the file name with cleaned MSKEYVALUEs.

Related Information

- Creating a New Package [page 17]
- Defining the Package Constant TUTORIAL_TARGET [page 18]
- Creating Repository Type for the Target Folder [page 19]
- Creating Repository for the Target Folder [page 21]
- Specifying the System Log Level [page 22]
- Disabling Automatic Attribute Creation [page 23]
- Creating Package JScript CreateFileName [page 24]
3.1 Creating a New Package

You create a package that contains all the configurations for this tutorial.

Context

Procedure

1. In the Identity Management Developer Studio tree view, expand your identity store, select the Packages folder and choose New Package from the context menu.

2. Enter the name of the new package. For example, tutorial.contexts
3. Choose OK.

3.2 Defining the Package Constant TUTORIAL_TARGET

You create a package constant to be able to reference the files created in this tutorial in a uniform way. The value of this constant is the path to the directory where the target repository for the files (FactorySystem folder) is to be placed.

Prerequisites

The package is checked out and ready for editing.

Context

Procedure

1. In the tree view, expand the test.tutorial.contexts package, select Constants and choose New Package Constant from the context menu.
2. Specify the name of the package constant, for example `TUTORIAL_TARGET`, and the directory where the folders are to be stored. Make sure that the directory actually exists (that is, the folders `Tutorial` and `Target` exist. If not, create them).

![Package Constant](image)

3. Choose OK.

### 3.3 Creating Repository Type for the Target Folder

You create a repository type for the target folder.

**Prerequisites**

The package is checked out and ready for editing.

**Context**

**Procedure**

1. In the tree view, expand the `test.tutorialcontexts` package, select `Repository Types` and choose `New Repository Type` from the context menu.
2. In the Repository Type dialog, specify the following:

Table 7:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository Type Template</td>
<td>Select Generic.</td>
</tr>
<tr>
<td>Repository Type Name</td>
<td>Enter the name of this repository type, for example FACTORY_SYSTEM</td>
</tr>
</tbody>
</table>

Choose OK.

3. Select the Assignment Grouping tab and make sure that assignment grouping is disabled for this repository type. That is, the No Grouping option should be selected on the Assignment Grouping tab of the FACTORY_SYSTEM repository type.

4. Select the General tab and add a new constant to the repository type definition by choosing New from the context menu. In the Repository Type Properties dialog, specify the following:

Table 8:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter PATH.</td>
</tr>
<tr>
<td>Category</td>
<td>Select Repository Type Constant With Override.</td>
</tr>
<tr>
<td>Data Type</td>
<td>Select String.</td>
</tr>
<tr>
<td>Value</td>
<td>Enter %$pck.TUTORIAL_TARGET\FactorySystem</td>
</tr>
<tr>
<td>Description</td>
<td>(Optionally) Enter a description.</td>
</tr>
</tbody>
</table>

Choose OK.
5. Save your changes.

### 3.4 Creating Repository for the Target Folder

You create a repository for the target folder.

**Context**

**Procedure**

1. Create a folder where users will be provisioned to. The folder will function as a target repository for the provisioning data. You need to create the folder `FactorySystem` in `C:\Tutorial\Target` (the directory you created a package constant for).
3. Select the `System configuration` tab and then `Repositories` from the menu on the left.
5. In the Create Repository dialog, enter a unique name for the new repository, for example: FACTORY_SYSTEM. Select the repository type for this repository FACTORY_SYSTEM.

6. Choose OK.

### 3.5 Specifying the System Log Level

To be able to view the log information in this tutorial, you set the log level for the system log to Info.

#### Context

#### Procedure

1. In the tree view, open the Identity Center properties.
2. Select the General tab.
3. In the Logging section, select Info to be the value of the System Log Level.

4. Save your changes.

3.6 Disabling Automatic Attribute Creation

You make sure that the option for the automatic attribute creation on the identity store is disabled.

Context

The Automatically Create Attributes option is used to control what happens when an attribute which does not exist or an attribute which is not defined as a legal attribute on an entry type is written to the identity store. If the Automatically Create Attributes option is enabled, the new attribute is created and added to the entry type. If the option is disabled, an error is returned.

Procedure

1. In the tree view, select your identity store and choose Properties from the context menu.
2. On the General tab, make sure the Automatically Create Attributes checkbox is not selected.
3. Save your changes.

3.7 Creating Package JScript CreateFileName

You create a package Java script that is used by the provisioning tasks to construct the file name using the cleaned MSKEYVALUEs of the privilege and context assigned to the user.

Prerequisites

The package is checked out and ready for editing.

Context

Cleaned MSKEYVALUE is MSKEYVALUE where the colon ("\:" ) is replaced by the underscore ("\_\:" ). The purpose is to make sure that it does not contain characters which are not allowed in a file name (not possible to use the colon ("\:" ) in a file name). The file name has the following naming convention: <cleaned MSKEYVALUE of the context>-<cleaned MSKEYVALUE of the privilege>-<MSKEYVALUE of the provisioned user>.txt
To create a package script, proceed as follows:

**Procedure**

1. In the tree view, expand the `test.tutorial.contexts` package, select `Scripts` and choose `New Package Script` from the context menu.

2. Enter script name, for example `CreateFileName`, and choose `OK`.

3. Define the following script (you can copy and paste the script defined below and replace the template definition):

```
// Main function: CreateFileName
function CreateFileName(Par){
    var SQL;
    var SQLResult;
    var Result;
    var CtxMSKEYVALUE;
    var ReturnValue;
    var CtxMSKEY;
    var IdS = uGetIDStore();
    SQL = "SELECT mcContextMSKEY, mcThisMSKEYVALUE, mcOtherMSKEYVALUE FROM idmv_link_ext WHERE mcUniqueID = " + Par;
    SQLResult = uSelect(SQL);
    Result = SQLResult.split("|");
    CtxMSKEY = Result[0];
    if (CtxMSKEY.equals("null")) {
        CtxMSKEYVALUE = "null";
    } else {
        CtxMSKEYVALUE = uIS_GetValue(CtxMSKEY, IdS, "MSKEYVALUE");
    }
    // --- Return string, which will return a filename with cleaned MSKEYVALUEs of context and privilege
    ReturnValue = uReplaceString(ReturnValue, ":", ",\_\_\_");
    return ReturnValue;
}
```
4. Save your changes.

```
// Main function: CreateFileName
function CreateFileName(Par)
{
    var SQL;
    var SQLResult;
    var Result;
    var OneValue;
    var ReturnValue;
    var CtxNSKEY;
    var SQL = uGetIDStore();
    SQL = "SELECT woContextNSKEY, woThisNSKEYVALUE, woOtherNSKEYVALUE FROM iadm_files WHERE woUniqueID = " + Par;
    SQLResult = uSelect(SQL);
    Result = SQLResult.splice("!");
    CtxNSKEY = Result[0];
    if (CtxNSKEY == null)
    {
        CtxNSKEYVALUE = uID_GetValue(CtxNSKEY, ID5, "NSKEYVALUE");
    } else {
        CtxNSKEYVALUE = uID_GetValue(CtxNSKEY, ID5, "NSKEYVALUE");
    }
// --- Return string, which will return a filename with cleaned NSKEYVALUES of
// context and privilege
return ReturnValue;
}
```
4 Building the Identity Store

In this section, you will learn how to use and populate the default identity store.

Related Information

- Displaying the Details for Entries [page 27]
- Reading the HR Data into the Identity Store [page 28]
- Creating a Context Type [page 33]
- Adding Contexts, Roles and Privileges [page 36]

4.1 Displaying the Details for Entries

To display the details for an entry in the Identity Management User Interface (on the Manage tab), you import the default forms package (com.sap.idm.forms.default) into your identity store.

Context

Procedure

1. In the tree view, select the identity store where you want to import the package to.
2. Select the Packages folder and choose Import from the context menu.
3. Navigate to the <Install_folder>\Core\ConfigurationPackages\Forms folder, select the com.sap.idm.forms.default.idmpck file and choose Open.
4. Provide the necessary information in the Enter import reason dialog box and choose OK.
4.2 Reading the HR Data into the Identity Store

In this section, the content of the sample database table `HR_Sample` is read into the identity store by creating a job which will read the source data to the identity store.

Related Information

- Creating the Job Folder and Adding the Job [page 28]
- Creating the Pass to Update the Identity Store [page 29]
- Running the Job [page 31]
- Verifying the Content of the Identity Store [page 32]

4.2.1 Creating the Job Folder and Adding the Job

You create a folder to hold the jobs in the tutorial, and then create the job definition for this job.

Prerequisites

The package is checked out and ready for editing.

Context

Procedure

1. In the tree view, select the `test.tutorial.contexts` package and then select the `Jobs` folder.
2. Choose `New > Job Folder`.
3. Rename the job folder, for example `Context Tutorial Jobs`.
4. Select your job folder and choose `New > Job` from the context menu.
5. Rename the job, for example `HR to Identity Store`. 
In case you do not have a default dispatcher on the system, select a dispatcher on the General tab of the job properties.

6. Save your changes.

Your next step is to add one pass to this job to read from the sample database table into the identity store.

4.2.2 Creating the Pass to Update the Identity Store

You create the pass that writes the data to the identity store.

Prerequisites

The package is checked out and ready for editing.

Context

Procedure

1. In the tree view, select the HR to Identity Store job and choose Open from the context menu.
2. In the job view, select the Passes tab and choose New To Identity Store pass from the context menu.
3. Rename the pass, for example HR to IDS.
4. Select the Source tab of the pass and configure the database table to be used.
Table 9:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Select the Database option and insert the system parameter <code>%ddm.identitycenter%</code>.</td>
</tr>
<tr>
<td>SQL Query</td>
<td>Enter the SQL statement to select all rows from the table <code>select * from HR_Sample</code>.</td>
</tr>
</tbody>
</table>

5. On the Destination tab, configure the following:

Table 10:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identity Store</td>
<td>Make sure <code>--Self--</code> is selected.</td>
</tr>
<tr>
<td>Entry Type</td>
<td>Select the entry type <code>MX_PERSON</code>.</td>
</tr>
<tr>
<td>Insert Template</td>
<td>Choose <code>Insert Template &gt; Data Source Template</code> to insert the definitions for the pass.</td>
</tr>
</tbody>
</table>
Modify the definition to use the attributes from the entry type. You can use the context menu to find the destination attributes.

6. Save your changes.

4.2.3 Running the Job

You run the job and then open the job log to verify that 50 entries are added.

Context

Procedure

1. In the tree view, select the HR to Identity Store job and choose Run from the context menu.
2. Open the job log and verify that the job added 50 entries to the identity store.
4.2.4 Verifying the Content of the Identity Store

The identity store contains all entries from the HR_Sample table. You verify this in the Identity Management User Interface.

Prerequisites

The Identity Management User Interface is installed and configured for your Identity Management system as described in Installing and Configuring SAP Identity Management User Interface.

Context

Procedure

   Provide the credentials of the user with access to the Manage tab.
2. Select the Manage tab.
3. In the Show field, select Person and choose Go.
4. Verify that the entries are present in the identity store.
4.3 Creating a Context Type

You create a context entry type.

Context

Procedure

1. In the tree view, expand the Identity Store Schema of your identity store.
2. Select Entry Types and choose New Entry Type from the context menu.
3. On the General tab, configure the following:

Table 11:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter a name for the new context type. For example, FACTORY</td>
</tr>
<tr>
<td>Context Entry Type</td>
<td>Enable the Context Entry Type option for this entry type.</td>
</tr>
<tr>
<td>Searchable Entry Type</td>
<td>Enable the Searchable Entry Type option for this entry type.</td>
</tr>
</tbody>
</table>
4. Select the **Attributes** tab.

5. Select **Mandatory** for the DISPLAYNAME attribute, and select **Allow** for the DESCRIPTION attribute. Make sure that **List** is selected for the MSKEYVALUE attribute.

   **Note**

   The MSKEYVALUE and MX_ENTRYTYPE attributes are mandatory and are added automatically as the entry type is saved.

6. Choose **OK** to save your changes.

7. Choose **No** when you are asked *Is this entry type used to hold identities?*. 
Now the created context type is displayed in the list of available entry types:

<table>
<thead>
<tr>
<th>Name</th>
<th>Add Entry Process</th>
<th>Modify Entry Process</th>
<th>Delete Entry Process</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FACTORY</td>
<td></td>
<td></td>
<td></td>
<td>Context: Tutorial, Context Entry Type</td>
</tr>
<tr>
<td>APPLICATION</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for application</td>
</tr>
<tr>
<td>APPLICATIONREQUEST</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for ASYN request</td>
</tr>
<tr>
<td>COMPANY_ADDRESS</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for company address</td>
</tr>
<tr>
<td>COMPANY_Group</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for company groups</td>
</tr>
<tr>
<td>GROUP</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for group</td>
</tr>
<tr>
<td>EMAIL</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email</td>
</tr>
<tr>
<td>EMAIL_ADDRESS</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email address</td>
</tr>
<tr>
<td>EMAIL_SUBJECT</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email subject</td>
</tr>
<tr>
<td>EMAIL_TO</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email to</td>
</tr>
<tr>
<td>EMAIL_CC</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email cc</td>
</tr>
<tr>
<td>EMAIL_BCC</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email bcc</td>
</tr>
<tr>
<td>EMAIL_ATTACHMENT</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email attachment</td>
</tr>
<tr>
<td>EMAIL_MESSAGE</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email message</td>
</tr>
<tr>
<td>EMAIL_SENDER</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email sender</td>
</tr>
<tr>
<td>EMAIL_READ</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email read</td>
</tr>
<tr>
<td>EMAIL_REPLY</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply</td>
</tr>
<tr>
<td>EMAIL_REPLY_TO</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply to</td>
</tr>
<tr>
<td>EMAIL_REPLY_CC</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply cc</td>
</tr>
<tr>
<td>EMAIL_REPLY_BCC</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply bcc</td>
</tr>
<tr>
<td>EMAIL_REPLY_ATTACHMENT</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply attachment</td>
</tr>
<tr>
<td>EMAIL_REPLY_MESSAGE</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply message</td>
</tr>
<tr>
<td>EMAIL_REPLY_SENDER</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply sender</td>
</tr>
<tr>
<td>EMAIL_REPLY_READ</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply read</td>
</tr>
<tr>
<td>EMAIL_REPLY_REPLY</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply reply</td>
</tr>
<tr>
<td>EMAIL_REPLY_REPLY_CC</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply reply cc</td>
</tr>
<tr>
<td>EMAIL_REPLY_REPLY_BCC</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply reply bcc</td>
</tr>
<tr>
<td>EMAIL_REPLY_REPLY_ATTACHMENT</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply reply attachment</td>
</tr>
<tr>
<td>EMAIL_REPLY_REPLY_MESSAGE</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply reply message</td>
</tr>
<tr>
<td>EMAIL_REPLY_REPLY_SENDER</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply reply sender</td>
</tr>
<tr>
<td>EMAIL_REPLY_REPLY_READ</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply reply read</td>
</tr>
<tr>
<td>EMAIL_REPLY_REPLY_REPLY</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply reply reply</td>
</tr>
<tr>
<td>EMAIL_REPLY_REPLY_REPLY_CC</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply reply reply cc</td>
</tr>
<tr>
<td>EMAIL_REPLY_REPLY_REPLY_BCC</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply reply reply bcc</td>
</tr>
<tr>
<td>EMAIL_REPLY_REPLY_REPLY_ATTACHMENT</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply reply reply attachment</td>
</tr>
<tr>
<td>EMAIL_REPLY_REPLY_REPLY_MESSAGE</td>
<td></td>
<td></td>
<td></td>
<td>Entry type for email reply reply reply message</td>
</tr>
</tbody>
</table>

8. Verify that the context type is displayed in the Identity Management User Interface. That is, verify that FACTORY is available in the list of entry types in the Show field on the Manage tab.

9. Repeat these steps and create the context type COUNTRY.
4.4 Adding Contexts, Roles and Privileges

You create a job that reads the content of the CSV file ContextBasedAssignments_tutorial.csv into the identity store. This creates the contexts, privileges, roles and the role and privilege hierarchy, as well as set the context type each role is allowed for, repository definition and the conditional context for the privileges.

Prerequisites

The package is checked out and ready for editing.

Context

Procedure

1. In the tree view, select the Context Tutorial Jobs job folder and choose New Job from the context menu.
2. Rename the job, for example: Load Tutorial Data.
3. Enable the job and select a dispatcher.
4. Save your changes.

Next Steps

Your next step is to add two passes to the newly created job Load Tutorial Data. The first pass will read the ASCII file ContextBasedAssignments_tutorial.csv into the temporary table tutorial_entries, and the second pass will read from this table into the identity store.

This should 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 CSV file. If the second pass is 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 entries from the identity store.
4.4.1 Reading the ASCII File

You create the pass that reads the ASCII file.

Context

Procedure

1. In the tree view, select the Load Tutorial Data job and open it in the job view.
2. Select the Passes tab and choose New From ASCII File pass from the context menu.
3. Rename the pass, for example: Read data.
4. Select the Source tab and configure the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
<td>Enter the path to the CSV file. For example: C:Users Administrator Desktop ContextBasedAssignments_tutorial.csv</td>
</tr>
<tr>
<td>Field Separator</td>
<td>Enter a comma sign (,) as the field separator.</td>
</tr>
<tr>
<td>Header Line</td>
<td>Make sure that Header Line is selected.</td>
</tr>
</tbody>
</table>
5. Select the **Destination** tab and configure the following:

Table 13:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Destination Database</strong></td>
<td>You can select either the <em>SAP Identity Management Database</em> option, or the <em>Database</em> option and insert the system parameter '%$ddm.identitycenter%' from the context menu.</td>
</tr>
<tr>
<td><strong>Table Name</strong></td>
<td>Enter <strong>tutorial_entries</strong> as the table name.</td>
</tr>
<tr>
<td><strong>Definitions</strong></td>
<td>Choose <strong>Insert Template</strong> and select <strong>Data Source Template</strong> to create the pass definitions.</td>
</tr>
</tbody>
</table>

**Note**

Do not use hyphen in table names, as this will cause problems with some database drivers.
6. Save your changes.

### 4.4.2 Running the Job

You run the **Load Tutorial Data** job to test the **Read data** pass.

#### Context

#### Procedure

1. In the tree view, select the **Load Tutorial Data** job and choose **Run** from the context menu.
2. View the job log to verify that the job ran successfully, and that a number of entries have been processed.
4.4.3 Updating the Identity Store

Your create the pass that writes the data to the identity store.

Context

Procedure

1. In the tree view, select the Load Tutorial Data job and open it in the job view.
2. Select the Passes tab and choose New To Identity Store pass from the context menu.
3. Rename the pass, for example: Add data to idstore.
4. Select the Source tab and configure the following:

Table 14:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Use the context menu to insert the system parameter $ddm.identitycenter%.</td>
</tr>
<tr>
<td>SQL Query</td>
<td>Enter the SQL statement to select all rows from the table created in the Read data pass select * from tutorial_entries.</td>
</tr>
</tbody>
</table>
5. Select the **Destination** tab and configure the following:

Table 15:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Identity Store</strong></td>
<td>Make sure that the --Self-- identity store is selected.</td>
</tr>
<tr>
<td><strong>Entry Type</strong></td>
<td>Do not specify an entry type. This will be dynamically loaded along with the other attributes.</td>
</tr>
<tr>
<td><strong>Definitions</strong></td>
<td>Modify the definition to use the attributes from the entry type. You can use the context menu to find the destination attributes.</td>
</tr>
</tbody>
</table>
6. Save your changes.

4.4.4 Running the Job and Verifying the Contents of Identity Store

You run the job and open the job log to verify that entries were processed and added.

Context

Procedure

1. In the tree view, select the Load Tutorial Data job and choose Run from the context menu.
2. View the job log to verify that the job ran successfully, and that a number of entries have been processed.

   The identity store should now contain all entries from the ContextBasedAssignments_tutorial.csv file.
Results

In the Identity Management User Interface, select the Manage tab and verify the following:

- These privileges exist: PRIV:Purchasing, PRIV:Distribution, PRIV:HRAdmin, PRIV:ProductionPlantNO, PRIV:ProductionPlantDE and PRIV:ProductionPlantFR.
- The repository definition FACTORY_SYSTEM is defined on the privileges. Use the Modify Privilege Details form to view these details.
- These roles and the role hierarchy exist: ROLE:Manager, ROLE:HR_Manager, ROLE:SupplyManager and ROLE:ProductionSupervisor.
- Inspect the roles to see that the context is defined for each role, and that the following member privileges are defined:
  - For ROLE:Manager – no privileges are defined
  - For ROLE:HR_Manager – PRIV:HRAdmin
  - For ROLE:SupplyManager – PRIV:Purchasing and PRIV:Distribution
  - For ROLE:ProductionSupervisor – PRIV:ProductionPlantNO PRIV:ProductionPlantDE and PRIV:ProductionPlantFR
- Inspect the privileges PRIV:ProductionPlantNO, PRIV:ProductionPlantDE and PRIV:ProductionPlantFR. Use the Modify Privilege Details form and then select the Context tab to see that the conditional context is defined for each of these.
  - For PRIV:ProductionPlantNO – conditional context COUNTRY:Norway
  - For PRIV:ProductionPlantDE – conditional context COUNTRY:Germany
  - For PRIV:ProductionPlantFR – conditional context COUNTRY:France
- Make sure that FACTORY:Norway, FACTORY:Germany and FACTORY:France are listed when you select FACTORY in the Show field.
- Make sure that COUNTRY:Norway, COUNTRY:Germany and COUNTRY:France are listed when you select COUNTRY in the Show field.
4.4.5 Viewing the Properties of the Tutorial Data

Reading the content of the `ContextBasedAssignments_tutorial.csv` file to the identity store provides you with most of the data necessary to complete this tutorial. This section describes the properties and the configuration of the content read into the identity store.

Related Information

- Roles and Privileges [page 44]
- Creating Context Entries [page 44]
- Defining Allowed Contexts on Privileges and Roles [page 47]
- Defining Conditional Contexts [page 48]

4.4.5.1 Roles and Privileges

For details about roles, privileges and their hierarchies, refer to Maintaining Identity Store Entries section in SAP Identity Management Configuration Guide.

Related Information

- Maintaining Identity Store Entries

4.4.5.2 Creating Context Entries

In this tutorial, you create context entries by reading a CSV file. There is another way of adding the context entries, that is, to create a form in the Identity Management Developer Studio that creates new entries of a given context type.

Context

In this tutorial you use two context types – FACTORY and COUNTRY, which means that two forms would need to be created, one for each context type.
To create a form that is used for creating entries of a given context type (for example, FACTORY), proceed as follows:

**Procedure**

1. In the tree view, select the **Forms** folder and choose **New → Form** from the context menu.
2. Rename the form. For example, **Create Contexts of Type FACTORY**.
3. Select the **Attributes** tab.
4. In the **Entry Type** field, select **FACTORY**.

5. Select the **Access Control** tab.
6. Choose **Add** and configure the following in the **Access Control** dialog:

   **Table 16:**

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow Access for</td>
<td>Select Logged-in user or identity store entry.</td>
</tr>
<tr>
<td>Name</td>
<td>Select Administrator.</td>
</tr>
<tr>
<td>On Behalf of</td>
<td>Select Everybody.</td>
</tr>
</tbody>
</table>

   ![Image of the Access Control tab configuration screen.]

---

*SAP Identity Management Tutorial: Context-Based Assignments*

*Building the Identity Store*
The access control allows the administrator user to run this form for all users.

7. Repeat these steps to create a form that is used to create entries of context type COUNTRY.

Results

Once created, the two forms can be used in the Identity Management User Interface to create entries of context types FACTORY and COUNTRY:

You can also add context entries by using a job with a To Identity store pass, and the following configuration (on the Destination tab):

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSKEYVALUE</td>
<td>FACTORY:France</td>
</tr>
<tr>
<td>DISPLAYNAME</td>
<td>FACTORY:France</td>
</tr>
<tr>
<td>changeType</td>
<td>add</td>
</tr>
</tbody>
</table>
4.4.5.3  Defining Allowed Contexts on Privileges and Roles

After the privileges, roles and their hierarchies are created in the identity store, the allowed contexts for each privilege and roles need to be defined.

Context

The allowed contexts for privileges and roles are visible in the Identity Management User Interface.

For privileges the option *Inherited from repository* is deselected.

**Note**

The allowed context type can be defined on the repository definition, and the privilege can inherit this allowed context type from the repository definition defined for each privilege. In this tutorial, there are two context types, and some privileges have one context type as the allowed context and some have the other context type as the allowed context. That is the reason why the allowed contexts are not inherited from the repository definition in this tutorial.

Allowed context for the privilege or role is selected from the list of context types. The allowed contexts for the privileges are defined in the following way:

<table>
<thead>
<tr>
<th>Privilege</th>
<th>Allowed Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRIV:HRAdmin</td>
<td>FACTORY</td>
</tr>
<tr>
<td>PRIV:Distribution</td>
<td>FACTORY</td>
</tr>
<tr>
<td>PRIV:Purchasing</td>
<td>FACTORY</td>
</tr>
<tr>
<td>PRIV:ProductionPlantDE</td>
<td>COUNTRY</td>
</tr>
<tr>
<td>PRIV:ProductionPlantFR</td>
<td>COUNTRY</td>
</tr>
<tr>
<td>PRIV:ProductionPlantNO</td>
<td>COUNTRY</td>
</tr>
</tbody>
</table>

The allowed contexts for the roles are defined in the following way:

<table>
<thead>
<tr>
<th>Role</th>
<th>Allowed Context</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROLE:Manager</td>
<td>FACTORY</td>
</tr>
<tr>
<td>ROLE:HR_Manager</td>
<td>FACTORY</td>
</tr>
<tr>
<td>ROLE:SupplyManager</td>
<td>FACTORY</td>
</tr>
</tbody>
</table>
To view the allowed context defined for the PRIV:HRAdmin, proceed as follows:

**Procedure**

1. In the Identity Management User Interface, select the *Manage* tab and then select *Privilege* in the *Show* field.
2. Search for the PRIV:HRAdmin privilege, select it and choose *Choose Task*.
3. Select *Modify Privilege Details > Choose Task > Context*.
   
   You see that the allowed context for this privilege is *FACTORY*.

### 4.4.5.4 Defining Conditional Contexts

For provisioning using conditional context (use case 3), the conditional contexts are defined for the privileges.

**Context**

There are three privileges for this use case, all defined for ROLE:ProductionSupervisor, and their conditional contexts are the following:

- PRIV:ProductionPlantDE – conditional context COUNTRY:Germany
- PRIV:ProductionPlantFR – conditional context COUNTRY:France
- PRIV:ProductionPlantNO – conditional context COUNTRY:Norway
To define the conditional context for a privilege (for example, PRIV:ProductionPlantDE), proceed as follows:

**Procedure**

1. In the Identity Management User Interface, select the **Manage** tab and then select **Privilege** in the **Show** field.
2. Search for the PRIV:ProductionPlantDE privilege, select it and choose **Choose Task**.
3. Select **Modify Privilege Details** ➤ **Choose Task** ➤ **Context**.

You see that the allowed context for this privilege is **COUNTRY**.

4. To add the conditional context, in the **Mandatory Contexts** pane, search for the allowed context, select the line to enable the **Add** button and then choose **Add**.

![Modify Privilege Details](image)

### 4.5 Creating Assignment Request Forms

You create the two assignment request forms used in the three use cases.

**Context**
Procedure

1. Create a folder that will be used for the assignment request forms in the current tutorial. In the tree view, select the *Forms* folder and then choose [New > Folder].
2. Rename the folder, for example *Context Tutorial Assignment Forms*, and save your changes.
3. Select the folder and choose [New > Assignment Request] form from the context menu.
4. Rename the folder, for example *Context-Based Assignment Use Case 1 and 2*.
5. Select the *Parameters* tab and configure the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entry type</td>
<td>Select the entry type this form is available for. Here you select MX_PERSON entry type.</td>
</tr>
<tr>
<td>Context is Mandatory</td>
<td>Enable this option if you want the context to be mandatory.</td>
</tr>
<tr>
<td>Context Type</td>
<td>You can select the following:</td>
</tr>
<tr>
<td></td>
<td>○ Context type</td>
</tr>
<tr>
<td></td>
<td>○ <em>None</em> if there is no context (that is, a regular assignment request).</td>
</tr>
<tr>
<td></td>
<td>○ <em>User select</em> if you want to allow user to choose the context types (if more than one context type available)</td>
</tr>
<tr>
<td></td>
<td>Here you select the context type <em>FACTORY</em>.</td>
</tr>
<tr>
<td>Multiselect Context</td>
<td>Enable this option to be able to multi-select contexts.</td>
</tr>
<tr>
<td>Reference Type</td>
<td>You can select MX_ROLE or MX_PRIVILEGE as the entry type this assignment request is assigning to the selected entry type (MX_PERSON). Here you assign a role to a user, that is, you select MX_ROLE as a reference type.</td>
</tr>
<tr>
<td>Multiselect Reference</td>
<td>Enable this option to be able to multi-select references.</td>
</tr>
<tr>
<td>Ask for Validity</td>
<td>You can select <em>Optional</em> or <em>Never</em>.</td>
</tr>
<tr>
<td></td>
<td>Here you select <em>Optional</em>.</td>
</tr>
<tr>
<td>Ask for Reason</td>
<td>You can select <em>Never</em>, <em>Mandatory</em> or <em>Optional</em>.</td>
</tr>
<tr>
<td></td>
<td>Here you select <em>Optional</em>.</td>
</tr>
</tbody>
</table>
6. Select the **Access Control** tab, choose **Add** and fill in the following in the **Access Control** dialog:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow access for</td>
<td>Select the logged-in user access type, that is, <em>Logged-in user or identity store entry</em>.</td>
</tr>
<tr>
<td>Name</td>
<td>Enter the name of the identity store user with the access to the <strong>Manage</strong> tab in the Identity management User Interface (here Administrator). You might use <strong>Check Name</strong> to ensure that the name you entered is correct and exists. This allows the administrator user to create new roles.</td>
</tr>
<tr>
<td>On Behalf of</td>
<td>Make sure that <em>Everybody</em> is selected.</td>
</tr>
</tbody>
</table>
7. Save your changes.

8. Perform these steps to create Context-Based Assignment Use Case 3 assignment request form. For this form, choose context type COUNTRY.
4.6 Creating Factory_AddEntry Process

You create provisioning process that provisions users to a system that does not understand the concept of contexts. You also define the process for the repository definition FACTORY_SYSTEM.

Context

Procedure

1. Create a folder that will be used for the processes in this tutorial. In the tree view, select the Processes folder and choose New Folder from the context menu.
2. Rename the folder, for example Factory Provisioning, and save your changes.
3. Select the Factory Provisioning folder and choose New Process from the context menu.
4. Rename the process, for example Factory_AddEntry.

The Factory_AddEntry process contains one action task Add File to FactorySystem Folder, which operates on the entry type MX_PENDING_VALUE and adds the file with the following naming convention <cleaned...>
MSKEYVALUE of the context>-<cleaned MSKEYVALUE of the privilege>-<MSKEYVALUE of the provisioned user>.txt to a specified directory (the FactorySystem folder). The contents of the file are date and time when the user is provisioned.

**Note**
This is given as an example only. There are no checks for illegal characters in the file name.

1. Select the General tab of the Factory_AddEntry process.
2. In the Repository field, select the FACTORY_SYSTEM repository.

**Results**

The process is now created and your next step is to add the Add File to FactorySystem Folder action task.

### 4.6.1 Adding the Task Add File to FactorySystem Folder

You add the Add File to FactorySystem Folder action task to the the Factory_AddEntry process.

**Context**

**Procedure**

1. In the tree view, select the Factory_AddEntry process and choose Open from the context menu.
2. Select **Action Task** from the palette to the right of the canvas.

3. Move the cursor to the connection where you want to add the task. When the connection is highlighted (yellow and dotted), click on it. The action task is automatically added and visible in the flow diagram.

4. Rename the action task, for example **Add File to FactorySystem Folder**.

5. Select the **Add File to FactorySystem Folder** action task and choose **Properties** from the context menu.

6. On the **General** tab, make sure the following is configured:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enabled</strong></td>
<td>Select this check box to enable the job to be run by a dispatcher.</td>
</tr>
<tr>
<td><strong>Run by Dispatchers</strong></td>
<td>Select a dispatcher that should be responsible for running this job.</td>
</tr>
</tbody>
</table>

7. In the job view, select the **Scripts** tab and choose **New** **Add Link to Package Script**. To establish the link to the **CreateFileName** package script, in the **Connection to Package Script** dialog, configure the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Select Package</strong></td>
<td>Make sure <strong>Self</strong> is selected.</td>
</tr>
<tr>
<td><strong>Select Script</strong></td>
<td>Select the package script <strong>CreateFileName</strong>.</td>
</tr>
</tbody>
</table>

8. Choose **OK**.
9. In the job view, select the **Passes** tab and choose **New** ➤ **Shell Execute** to create a pass.

10. Rename the pass, for example **Add File to FactorySystem Folder**.

11. Select the **Source** tab of the pass and make sure that **Retrieve Attributes From Pending Value** is selected.

12. Select the **Destination** tab of the pass and add the following line to the definitions (you can use the context menu to insert the constants/attributes/scripts or copy and paste the following lines): `cmd /c echo Privilege(s) assigned %ddm.date% %ddm.time% > "%$rep.PATH%\$FUNCTION.CreateFileName(%MX_LINK_REFERENCE%)$"`.

13. Save your changes.
4.6.2 Defining the Process on the Repository Type Definition

You define the *Factory_AddEntry* process to the FACTORY_SYSTEM repository type.

**Context**

**Procedure**

1. In the tree view, select the FACTORY_SYSTEM repository type and choose Properties from the context menu.
2. Select the Event Process tab.
3. In the Assignment section, choose ... to the right of the Add process. Navigate to and select the Factory_AddEntry process.
4. Save your changes.

**Results**

Now the link is defined on the FACTORY_SYSTEM repository type.
4.7 Creating Factory_RemoveEntry Process

You create deprovisioning process that is to deprovision users from a system that does not understand the concept of contexts.

Prerequisites

The default forms package `com.sap.idm.forms.default` is imported.

Context

Make sure to have a `Change Identity` form available in the Identity Management User Interface (by default this form is delivered with the `com.sap.idm.forms.default` package), where roles or privileges can be removed from the user to trigger the deprovisioning process.

The process `Factory_RemoveEntry` consists of one action task, `Delete File from FactorySystem Folder`, that deletes the previously created file from the folder.

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>This is given as an example only. There are no checks for illegal characters in the file name.</td>
</tr>
</tbody>
</table>

Procedure

2. Rename the process, for example `Factory_RemoveEntry`.
3. Select the `General` tab of the `Factory_RemoveEntry` process.
4. In the `Repository` field, select `FACTORY_SYSTEM` repository.
5. Save your changes.

Results

The process is now created and your next step is to add the `Delete File From FactorySystem Folder`. 
4.7.1 Adding the Task Delete File from FactorySystem Folder

You add the Delete File from FactorySystem Folder action task to the Factory_RemoveEntry process.

Context

Procedure

1. In the tree view, select the Factory_RemoveEntry process and choose Open from the context menu.

   Note

   If you just have created the process, the process flow diagram opens automatically when the process is created.

2. Select Action Task from the palette to the right of the canvas.
3. Move the cursor to the connection where you want to add the task. When the connection is highlighted (yellow and dotted), click on it. The action task is automatically added and visible in the flow diagram.
4. Rename the action task, for example Delete File from FactorySystem Folder.
5. Select the **Delete File from FactorySystem Folder** action task and choose **Properties** from the context menu.

6. On the **General** tab, make sure the following is configured:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enabled</strong></td>
<td>Select this check box to enable the job to be run by a dispatcher.</td>
</tr>
<tr>
<td><strong>Run by Dispatchers</strong></td>
<td>Select a dispatcher that should be responsible for running this job.</td>
</tr>
</tbody>
</table>

7. In the job view, select the **Scripts** tab and choose **New** \(\rightarrow\) **Add Link to Package Script**. To establish the link to the **CreateFileName** package script, in the **Connection to Package Script** dialog, configure the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Select Package</strong></td>
<td>Make sure <strong>Self</strong> is selected.</td>
</tr>
<tr>
<td><strong>Select Script</strong></td>
<td>Select the package script <strong>CreateFileName</strong>.</td>
</tr>
</tbody>
</table>

8. Choose **OK**.

9. In the job view, select the **Passes** tab and choose **New** \(\rightarrow\) **Shell Execute** to create a pass.

10. Rename the pass, for example **Delete File from FactorySystem Folder**.

11. Select the **Source** tab of the pass and make sure that **Retrieve Attributes From Pending Value** is selected.

12. Select the **Destination** tab of the pass and add the following line to the definitions (you can use the context menu to insert the constants/attributes/scripts or copy and paste the following lines): `cmd /c Del "%rep.PATH%\$FUNCTION.CreateFileName(%MX_LINK_REFERENCE%)\$"`. 
13. Save your changes.

4.7.2 Defining the Process on the Repository Type Definition

You define the Factory_RemoveEntry process to the FACTORY_SYSTEM repository type.

Context

Procedure

1. In the tree view, select the FACTORY_SYSTEM repository type and choose Properties from the context menu.
2. Select the Event Process tab.
3. In the Assignment section, choose ... to the right of the Add process. Navigate to and select the Factory_RemoveEntry process.
4. Save your changes.
Results

Now the link is defined on the FACTORY_SYSTEM repository definition.
5 Performing Basic Context-Based Provisioning (Use Case 1)

To perform a basic context-based provisioning of users, both context and a role (and thus privilege(s)) need to be selected in the assignment request form Context-Based Assignment Use Case 1 and 2.

Context

Procedure

1. In the Identity Management User Interface, select the Manage tab. Make sure that the Person is selected in the Show field and choose Go.
2. Select entry 3001 and choose Choose Task.
   Forms available for the entry type MX_PERSON are displayed in the User Interface forms folder. Expand the folder and select the Context-Based Assignment Use Case 1 and 2 form.
3. Choose Choose Task.
   The Context-Based Assignment Use Case 1 and 2 form opens in a new window. This assignment request form consists of four steps.
4. Select a context.
   a. View that the defined context type is FACTORY, so it is selected in the Context Type field.
   b. Choose Search. This lists all available contexts.
   c. Select a context, here the context FACTORY:Germany, and choose Next.

5. Select a role.
   a. Choose Go. This lists all available roles.
   b. Select the role ROLE:HR_Manager.
   c. Choose Add and then Next.

6. Enter details.
a. Define validity period and (optionally) a reason for the role.
b. If you leave the validity period empty, it will be set to Unlimited.

7. Review and finish.
   a. Summarize the assignment. Review the information and make changes if necessary.
   b. Choose Finish to save and execute the form.

Results

The role ROLE:HR_Manager is now assigned to user 3001 with the context FACTORY:Germany. Assigning ROLE:HR_Manager to user 3001, gives the user the privilege PRIV:HRAdmin.

Go to directory C:\Tutorial\Target\FactorySystem and verify that a file is created for the user 3001.

To deprovision the user, remove the role ROLE:HR_Manager from the user 3001 by using Change Identity form. Go to directory C:\Tutorial\Target\FactorySystem and verify that the file created for the user 3001 is now removed.
6 Performing Provisioning Using Default Context (Use Case 2)

Provisioning using default (or automatic) context is quite similar to the basic context-based provisioning. The difference is that the employee to be provisioned has a defined default context.

**Context**

This is done by defining the context as the value for the MX_CTX_AUTO_VALUES attribute on the person entry. The default context is automatically added during the assignment due to defined default context for the user. Autoassign context policies used in this use case are **If missing** and **Always**.

**Procedure**

1. In the Identity Management User Interface, use a form for modifying user properties (that is, a modify form that uses MX_PERSON entry type and specifies the MX_CTX_AUTO_VALUES attribute) to define the default context for a given user.

   Here, you define the **Auto-Assigned Contexts** (MX_CTX_AUTO_VALUES attribute) for user 3001 to be **FACTORY:Germany**.

2. In Identity Management Developer Studio, define the autoassign context policy to be **If Missing** on the repository type.
To do so, open the FACTORY_SYSTEM repository type properties, select the Context tab and define If Missing for Autoassign Context Policy field.

3. Provision the user (here, the user 3001) by choosing the Context-Based Assignment Use Case 1 and 2 form in the Identity Management User Interface. This assignment request form consists of four steps.

4. The default context for the user is defined. To demonstrate the autoassign context policy If Missing, select no context in the first step of the assignment request form, that is, proceed directly to step 2.

5. Select the role for the assignment, here ROLE:Manager.

6. Define validity period and (optionally) a reason for the role.

7. Summarize the assignment. Review the information and make changes if necessary.

Choose Finish to save and execute the form.
Assigning ROLE:Manager to the user 3001, gives the user the privilege PRIV:HRAdmin (inherited from ROLE:HR_Manager) and privileges PRIV:Purchasing and PRIV:Distribution (inherited from ROLE:SupplyManager).

8. Go to directory C:\Tutorial\Target\FactorySystem and verify the files created for the user 3001. Due to If Missing autoassign context policy, which will assign the default context only if no context is selected during the assignment, the user is provisioned with the defined default context (FACTORY:Germany).

9. Choose another user and define the default context (here you define default context FACTORY:France for user 3002).
10. Define the autoassign context policy to be *Always* on the FACTORY_SYSTEM repository type.

11. Provision the user 3002 using the *Context-Based Assignment Use Case 1 and 2* form. The default context for the user is defined. Select a context for the user in the first step of the form (you can select any context but the one that is defined as default for your user). Here you select context FACTORY:Norway. Then proceed to select the role for the assignment (here you select ROLE:SupplyManager).
Assigning ROLE:SupplyManager to the user 3002, gives the user the privileges PRIV:Purchasing and PRIV:Distribution.

12. Go to directory C:\Tutorial\Target\FactorySystem and verify the files created for the user 3002:

Due to autoassign context policy Always, which will always assign the default context unless already assigned, the user is provisioned with the following privileges and contexts:

- PRIV:Purchasing with context FACTORY:France (the default context)
- PRIV:Distribution with context FACTORY:France (the default context)
- PRIV:Purchasing with context FACTORY:Norway (context selected during assignment)
- PRIV:Distribution with context FACTORY:Norway (context selected during assignment)

To deprovision the user(s), remove the roles (here ROLE:Manager for user 3001 and/or ROLE:SupplyManager for user 3002) using the Change Identity form.
7 Performing Provisioning Using Conditional Context (Use Case 3)

The purpose of the conditional context is to assign only those privileges for which a given context is present. That is, when the user is assigned ROLE:ProductionSupervisor with a given context, then only the privilege that has a matching context is assigned to the user.

Context

If the context selected during the assignment is COUNTRY:France, then the user will only get assigned the privilege PRIV:ProductionPlantFR which has the context COUNTRY:France as the conditional context.

Procedure

1. Provision a user using the Context-Based Assignment Use Case 3 form. Here you provision the user 3003.

2. Select the context and the role for the user. Here you select context COUNTRY:France and the role ROLE:ProductionSupervisor (which is the only role available for this use case). Optionally enter the validity and reason in step 3 of the assignment request form, then review and choose Finish to run the form.
Assigning ROLE:ProductionSupervisor with the context COUNTRY:France to the user 3003, gives the user the privilege PRIV:ProductionPlantFR due to the conditional context.

3. Go to directory C:\Tutorial\Target\FactorySystem and verify the file created for the user 3003:

Next Steps

To deprovision the user, remove the ROLE:ProductionSupervisor role by using the Change Identity form.
8 Troubleshooting

If any problems occur during the execution, you check the suggested solutions.

- 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 all paths (for the target system and files) are correct.
- 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 on the **Logging** tab of the job properties. You can also deselect the **Reset log file** checkbox 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.
Important Disclaimers and Legal Information

Coding Samples
Any software coding and/or code lines / strings (“Code”) included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP’s gross negligence.

Accessibility
The information contained in the SAP documentation represents SAP’s current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP in particular disclaims any liability in relation to this document. This disclaimer, however, does not apply in cases of willful misconduct or gross negligence of SAP. Furthermore, this document does not result in any direct or indirect contractual obligations of SAP.

Gender-Neutral Language
As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with “you”, or a gender-neutral noun (such as “sales person” or “working days”) is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

Internet Hyperlinks
The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP’s gross negligence or willful misconduct. All links are categorized for transparency (see: http://help.sap.com/disclaimer).