cockpit
IT Service Manager

Ticketing - Synchronizing Cockpit ITSM with ServiceNow

FAQ document

Table of contents

Introduction

I. Objective

This goal of this document is to present a functional example of a synchronization between a Cockpit ITSM instance and a ServiceNow Jakarta instance.

II. Principles

- Please see the document "FAQ - Synchronization User Guide" for more information on the functions and features of the Cockpit ITSM menu.
- While reading this document, plus refer simultaneously to "FAQ - ServiceNow XML Source Code Synchronization," the source document that is explained in the present document.
- A firm understanding of the ServiceNow solution is required.

Configuration

Objectives: Before writing the XML code for synchronizing a Cockpit ITSM portal with a ServiceNow instance, the following programming elements must be taken into consideration.

I. Types of tickets

You can create a variety of different tickets in Cockpit ITSM and ServiceNow. In the table below, we have listed the types of tickets (with their corresponding names in each system) that we will be synchronizing.

Cockpit ITSM	ServiceNow
REQUEST	TASK
INCIDENT	INCIDENT
CHANGE REQUEST	CHANGE REQUEST

II. Statuses

Principle:

The statuses of the tickets will be mapped between Cockpit ITSM and ServiceNow.

In the example below, we have mapped the ServiceNow status "1" with the Cockpit ITSM status "OP_NEW":

```
<valueMap externalValue="1" cockpitValue="OP_NEW"/>
```

A. Cockpit ITSM

Menu: Tickets > Configuration > Incidents / Requests / Changes > Statuses

Principles:

- For custom statuses, use the value of the status's "Description" field.
- For standard statuses, the value to be used does not appear in the portal menu; you must use the values provided in the table below.

Status	Value to be used in the XML code
New	OP_NEW
To do	OP_TODO
Pending	OP_WAITING
In process	OP_PROCESS
Reopened	CL_REJECTED
Closed	CL_TEMP
Closed definitively	CL_DEF

B. ServiceNow

Principles:

- Not all ticket types have the same statuses.
- In the table below, the value from the "ID" column must be used in the XML code.

ID	Status	Table (type of ticket)
-5	New	change_request
-4	Assess	
-3	Authorized	
-2	Scheduled	
-1	Implement	
0	Review	
3	Closed	
4	Canceled	
1	New	incident
2	In Progress	
3	On Hold	
6	Resolved	
7	Closed	
8	Canceled	
-5	Pending	task
1	Open	
2	Work in Progress	
3	Closed Complete	
4	Closed Incomplete	
7	Closed Skipped	

III. Priorities

Principles:

The priorities of the tickets will be mapped between Cockpit ITSM and ServiceNow.

In the example below, we have mapped the ServiceNow priority "1" with the Cockpit ITSM priority "High":

```
<valueMap externalValue="1" cockpitValue="High"/>
```

A. Cockpit ITSM

When you create a ticket in Cockpit ITSM, 3 priorities can be selected regardless of the type of ticket (Request, Incident, or Change):

- High
- Medium

- Low

B. ServiceNow

When you create an incident or request in ServiceNow, the priority is characterized by the following fields:

- Impact (modifiable):
  - 1 – High
  - 2 – Medium
  - 3 – Low
- Urgency (modifiable):
  - 1 – High
  - 2 – Medium
  - 3 – Low
- Priorities: This field cannot be modified; it is the result of the combination of the two previous fields. The list of combinations results in 5 possible priorities:

Impact	Urgency	Priority
1 – High	1 – High	**1 – Critical**
1 – High	2 – Medium	**2 – High**
1 – High	3 – Low	**3 – Moderate**
2 – Medium	1 – High	2 – High
2 – Medium	2 – Medium	3 – Moderate
2 – Medium	3 – Low	**4 – Low**
3 – Low	1 – High	3 – Moderate
3 – Low	2 – Medium	4 – Low
3 – Low	3 – Low	**5 – Planning**

In the event of a synchronization of Cockpit ITSM => ServiceNow, you will be required to assign an impact and an urgency based on a Cockpit ITSM priority.

In the event of a synchronization of ServiceNow => Cockpit ITSM, you must map the 5 ServiceNow priorities with the 3 Cockpit ITSM priorities; as a result, certain ServiceNow priorities will be mapped with the same priorities in Cockpit ITSM, with the choice of priorities contingent upon the context.

> Please note: this does not apply for Change Request tickets as change requests are not prioritized in ServiceNow.

IV. User

Principle: A ServiceNow user is responsible for creating and updating tickets in ServiceNow after a synchronization. A ServiceNow user must be created to perform these actions.

Menu: In ServiceNow, go to "User Administration > Users" and create a new user.

Configuration:

- User ID: Choose an explicit "ID" (e.g. "Cockpit"); this ID will appear in the history of the changes which have been made to the ServiceNow tickets.
- "First name" and "Last name": Select a first and last name.
- Check the "Active" option.
- Time zone: GMT
- Roles: Select the "admin" role.

Retrieve the user's sys_id:

To create the tickets in ServiceNow, we need the “sys_id” of the ServiceNow user we have created previously. The “sys_id” will be fulfilled in the field “caller_id” of tickets.

To retrieve the “sys_id”:

- In ServiceNow go to the menu “User Administration > Users”
- Edit the user dedicated to the synchronisation with Cockpit ITSM
- Click on the hamburger menu “Additional actions” and then click on “Copy sys_id”
- In the XML file, paste the “sys_id” in the following part:

```
<set externalField="caller_id" cockpitField="NA" value="8ebf28a6dba01300d2c85421cf9619e6" direction="OUT"/>
```

V. Tables and fields

Objective: It may be necessary to search for information in the ServiceNow tables and fields.

Menu: In ServiceNow, go to:

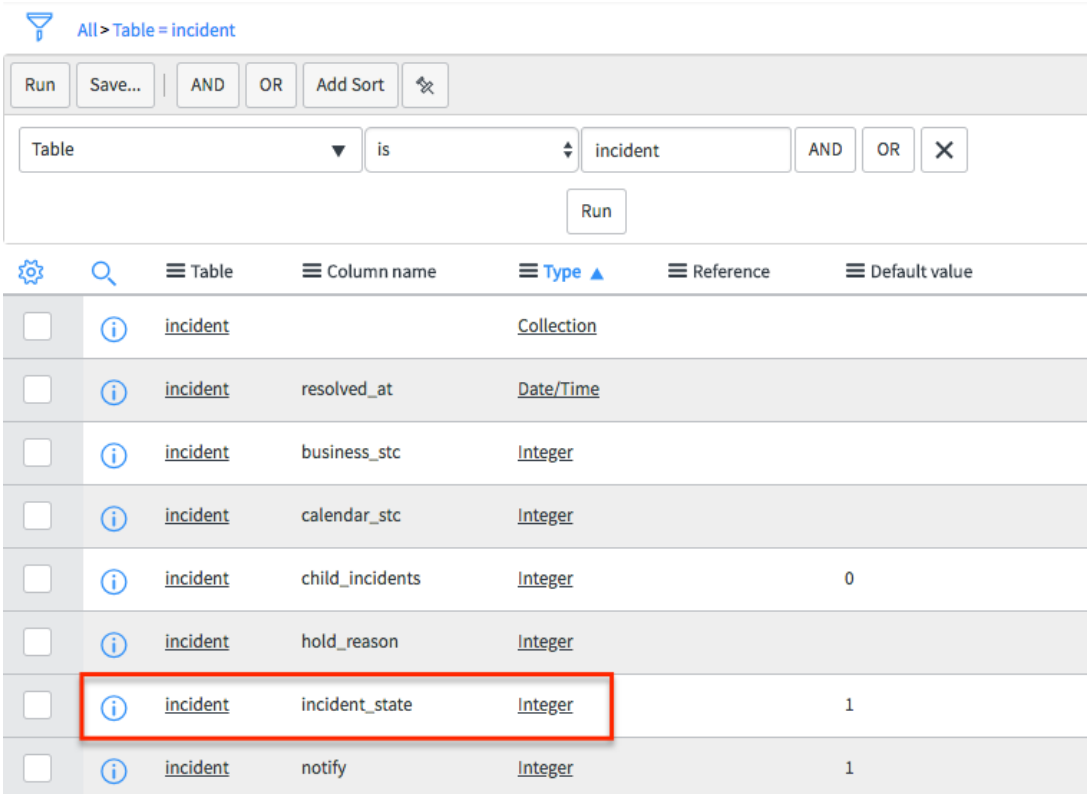
- System Definition > Dictionary: Allows you to search tables or fields with various filters.
- System Definition > Tables: Lets you search tables.

Example:

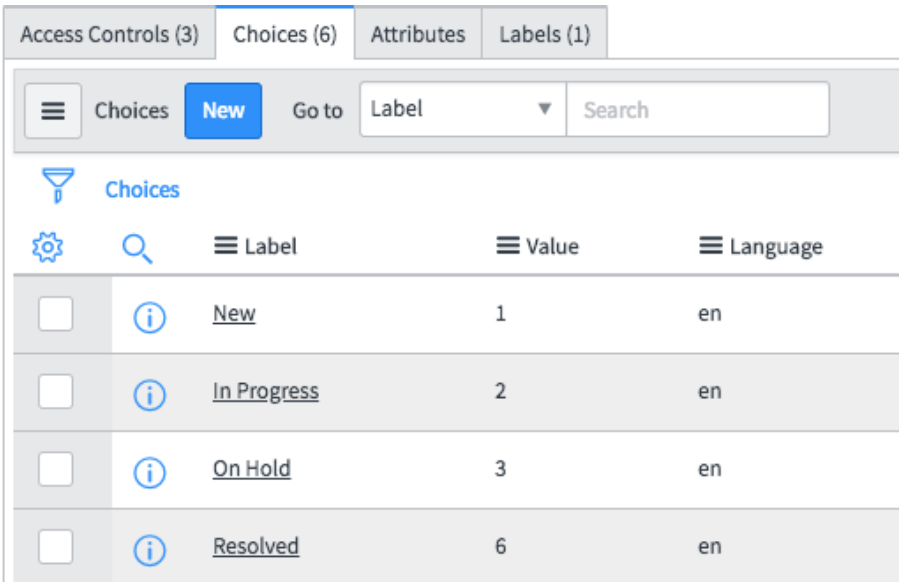
Verifying the statuses that are available in the incident table

- Go to the "System Definition > Dictionary" menu and search for the "incident" table.

- Click on the "incident_state" line.



- In the next window, click on the "Choices" tab:



XML source file

Objectives:

- Provide a concrete example of an XML file which will synchronize a Cockpit ITSM instance with a ServiceNow Jakarta instance.
- The example should serve as a basis for synchronizations with other ServiceNow instances, allowing you to adapt the source file to your own particular context.

I. <connector> tag

```
<?xml version="1.0" encoding="UTF-8"?>
<ticketSync xmlns="http://www.cockpit-itsm.com/TicketSyncConfiguration" description="Service Now Jakarta" syncFrequencyInMinutes="5">
```

- Do not modify the "xmlns" attribute value.
- "**syncFrequencyInMinutes**" indicates the frequency of the synchronization in minutes.
- "**allowProcessChange**" should be set to "true" if you would like for a ticket type to be synchronized in the event that the ticket type is changed.

```
    <connector id="service_now" username="cockpit" password="xxxxxxxx" url="https://devxxxxx.service-now.com" defaultTimeZone="UTC" dateFormat="yyyy-MM-dd HH:mm:ss">
```

- **Start of the <connector> tag.**
- "**id**" specifies the type of connector you will be using. For ServiceNow, the connector type is "service_now".
- "**username**" and "**password**" are the IDs of the ServiceNow user which was created in the previous section, and which is used by the connector to connect to the ServiceNow instance.
- "**url**" is the URL that is used to access the ServiceNow instance.
- "**defaultTimeZone**": Default time zone used to interpret the date fields of the ServiceNow instance. The time zone of a date field is not detected during the synchronization of the field. To avoid confusion, simply enter "UTC". The Cockpit ITSM user in charge of creating tickets originating from ServiceNow will also set these tickets to UTC.

```
    <parameters>
      <entry>
        <string>traceDirectory</string>
        <string>/tmp/ServiceNow</string>
      </entry>
```

- The log directory is found on the machine that hosts the portal. If you are in SaaS mode, you cannot access this element and must therefore remove the <entry /> tag.
- If you are in Premium mode, replace the "/tmp/ServiceNow" path with the appropriate path.

```
<entry>
  <string>dateFields</string>
  <string>
    // common fields,
    activity_due,
    approval_set,
    business_duration,
    calendar_duration,
    closed_at,
    due_date,
    expected_start,
    follow_up,
    opened_at,
    sla_due,
    sys_created_on,
    sys_updated_on,
    work_end,
    work_start,

    // incident fields,
    incident.resolved_at,

    // change_request fields,
    change_request.cab_date,
    change_request.conflict_last_run,
    change_request.end_date,
    change_request.requested_by_date,
    change_request.review_date,
    change_request.start_date,

    // request fields,
    sc_req_item.cab_date,
    sc_req_item.estimated_delivery
  </string>
</entry>
```

- By default, all ServiceNow "Date" fields are sent in a "String" format. In order for these fields to appear as dates, they must be identified and then defined as above.
- The fields listed above:
  - either belong to one table (e.g. "Incident.resolved_at");
  - or are present in several tables.
- To search other date fields, go to the "System Definition/Dictionary" menu and filter by field type (Date, Date/Time, Due date, etc.), table name, etc.

```
<entry>
  <string>readOnlyFields</string>
  <string>
    __type__,
    number
```

```
        </string>
      </entry>
```

- Allows you to disregard the "**type**" and "**number**" fields (ID of the ticket in ServiceNow) in read-only mode in the update processes that are sent to ServiceNow.

```
      <entry>
        <string>incomingDataFilter.change_request</string>
        <string>data.__operation__ != 'CREATE' || data.state === undefined || data.state === null ||
(data.state &amp;gt; -3 &amp;amp;&amp;amp; data.state &amp;lt; 4)</string>
      </entry>
```

- The **incomingDataFilter.change_request** parameter specifies a filter to be used for the incoming data of the ServiceNow "change_request" table. Any data that does not pass through this filter is ignored.
- Here, the filter is defined inside a JavaScript expression. The filter ensures that only update operations and the change request statuses "-3" and "4" are taken into account.

```
      <entry>
        <string>includeCreationDataFilter.change_request</string>
        <string>data.state == -2</string>
      </entry>
    </parameters>
  </connector>
```

- The **includeCreationDataFilter.change_request** parameter is a filter that is applied to the data which has passed through the previous "incomingDataFilter.change_request" filter.
- This filter is also defined as a JavaScript expression. In the previous filter, we exclude the ticket creations. In this filter, an "update" action with a "-2" status ("scheduled" status in ServiceNow) triggers the creation of a ticket.
- These 2 filters are used to take into account the fact that one ServiceNow change request may correspond to several Cockpit ITSM tickets.

> If you do not wish to synchronize the "Change Request" tickets, you can ignore the previous 2 filters.

- **End of the <connector> tag.**

II. The timestamping of actions

```
  <lastUpdateField>sys_updated_on</lastUpdateField>
```

- Map the dates of the operations on the external tickets to maintain the chronological order of the operations on both sides (Cockpit ITSM and ServiceNow).
- The "sys_updated_on" field can be found in every ServiceNow ticket.

> Note: "sys_update_on" is one of the "Date" fields that are previously defined in the XML file.

III. Synchronization process - Incident

- In the following example, we will describe the process for synchronizing "Incident" tickets.
- The synchronization will be done in both directions (Cockpit ITSM => ServiceNow and vice versa).
- The process treats creations, updates, and closures separately, thereby making it possible to select the fields you wish to synchronize according to the type of action.

A. Ticket type

```
<process ticketType="INCIDENT" discriminatorField="__type__" discriminatorValue="INCIDENT">
```

- The synchronization process begins with the <process /> tag.
- "discriminatorField" **attribute**: the "__type__" value corresponds to the type of ticket
- "discriminatorValue" **attribute**: the field value corresponds to the INCIDENT process

B. Mapping the IDs of tickets

```
<ticketIdMap externalField="number" cockpitField="ticket.externalReference"/>
```

- The <ticketIdMap /> tag maps ServiceNow tickets with Cockpit ITSM tickets:
  - "externalField" **attribute**: contains the value of the "number" field, which corresponds to the ID field of the ServiceNow ticket.
  - "cockpitField" **attribute**: contains the ID of the Cockpit ITSM ticket.
- If a ticket ID is not associated with another ID, then the operation is considered a creation.
- When a ticket ID is associated with another ID, then the operation is considered an update or a closure, depending on the status.

C. Creating a ticket

```
<create direction="BOTH" attachmentPrivacy="ALL">
```

- The <create /> tag is used to create tickets.
- If the "**BOTH**" value appears in the "direction" attribute, this means that the tickets created in ServiceNow will also be created in Cockpit, and vice versa. It is therefore possible to synchronize both systems in just one direction by using the values "IN" and "OUT".

```
<filters>
  <filter direction="IN">
    <filterRule field="__operation__" value="CREATE"/>
    <filterRule field="sys_created_by" value="cockpit" inverted="true"/>
  </filter>
</filters>
```

- First of all, apply a filter to the incoming tickets (the "IN" in the "direction" attribute) to ignore the ServiceNow tickets created by the "cockpit" user, i.e. the ServiceNow user in charge of creating tickets (see "Prerequisite").
  Without this filter, you risk entering into an infinite loop.
- It is also possible to filter the ServiceNow field "assignment_group" such that you only synchronize those tickets which are created by certain teams. Example:

```
<filterRule field="assignment_group" value="Team_Name"/>
```

1. Mapping of ServiceNow statuses and priorities => Cockpit ITSM

```
<mappers>
  <copy externalField="number" cockpitField="ticket.externalReference" direction="IN"/>
  <lookup targetType="status" externalField="state" cockpitField="ticket.status" direction="IN">
    <valueMap externalValue="1" cockpitValue="OP_NEW"/>
    <valueMap externalValue="2" cockpitValue="OP_PROCESS"/>
    <valueMap externalValue="3" cockpitValue="OP_WAITING"/>
    <valueMap externalValue="6" cockpitValue="CL_TEMP"/>
    <valueMap externalValue="7" cockpitValue="CL_DEF"/>
    <valueMap externalValue="8" cockpitValue="CL_REJECTED"/>
    <valueMap externalValue="*" cockpitValue="OP_NEW"/>
  </lookup>
```

- The <mappers> tag is used to indicate that the mapping of the fields has begun. All ticket fields that are not mapped will be ignored.
- A **direction** attribute with a value of "IN" means we will start by synchronizing the tickets in the following direction: ServiceNow => Cockpit. All subsequent "direction" attributes also have the value "**IN**".
- We start by mapping the ticket IDs with the **externalField** and **cockpitField** attributes.
- We use the <lookup> tag to map those fields which have predefined values (status, environment, etc.). The **targetType**, **externalField**, and **cockpitField** attributes map the statuses.

> Note: Use a <valueMap> tag with a value of "*" to ensure that a ticket is created even if a ServiceNow status is not provided in the mapping.

```
<copy externalField="short_description" cockpitField="ticket.title" stripHtml="true" direction="IN"/>
<copy externalField="description" cockpitField="ticket.request" stripHtml="true" direction="IN"/>
<copy externalField="opened_at" cockpitField="ticket.creationDate" direction="IN"/>
```

- The <copy> tag is used to map fields without modifying their contents. Example: text fields such as "ticket title".
- The "stripHtml" attribute is used to conserve the HTML formatting of the text.

```
<lookup targetType="priority" externalField="priority" cockpitField="ticket.priority" direction="IN">
  <valueMap externalValue="1" cockpitValue="High"/>
```

- Here, <lookup> is used to map the priorities of ServiceNow tickets with the priorities of Cockpit ITSM. To synchronize all the tickets, all the priorities must be taken into consideration. In Cockpit ITSM, the choice of priorities will be contingent upon the context.

> Note: When synchronizing in the direction ServiceNow => Cockpit ITSM, it is not necessary to take the "urgency" into account as this notion does not exist in Cockpit ITSM. This field is therefore not mapped.

```
        <lookup targetType="team" externalField="assignedTeam" cockpitField="ticket.assignedTeam"
direction="IN">
            <valueMap externalValue="*" cockpitValue="DEVELOPMENT"/>
        </lookup>
```

- The **targetType** attribute lets you assign a team to the ticket which has been created (the DEVELOPMENT team in the example).

2. Mapping of Cockpit ITSM statuses and priorities => ServiceNow

```
        <set externalField="__type__" cockpitField="N/A" value="INCIDENT" direction="OUT"/>
        <lookup targetType="status" externalField="state" cockpitField="ticket.status" direction="OUT">
            <valueMap cockpitValue="OP_NEW" externalValue="1"/>
            <valueMap cockpitValue="OP_TODO" externalValue="1"/>
            <valueMap cockpitValue="OP_PROCESS" externalValue="2"/>
            <valueMap cockpitValue="OP_WAITING" externalValue="3"/>
            <valueMap cockpitValue="CL_TEMP" externalValue="6"/>
            <valueMap cockpitValue="CL_DEF" externalValue="7"/>
            <valueMap cockpitValue="CL_REJECTED" externalValue="8"/>
            <valueMap cockpitValue="*" externalValue="1"/>
        </lookup>
        <copy externalField="short_description" cockpitField="ticket.title" stripHtml="true"
direction="OUT"/>
        <copy externalField="description" cockpitField="ticket.request" stripHtml="true" direction="OUT"/
>
        <!-- creation date is a Timestamp object. Leave it as is to let the ServiceNow user do the
conversion -->
        <copy externalField="opened_at" cockpitField="ticket.creationDate" targetClass="java.util.Date"
direction="OUT"/>
        <set externalField="caller_id" cockpitField="NA" value="8ebf28a6dba01300d2c85421cf9619e6"
direction="OUT"/>
        <lookup targetType="priority" externalField="urgency" cockpitField="ticket.priority"
direction="OUT">
            <valueMap externalValue="1" cockpitValue="High"/>
            <valueMap externalValue="2" cockpitValue="Medium"/>
            <valueMap externalValue="3" cockpitValue="Low"/>
        </lookup>
        <lookup targetType="priority" externalField="impact" cockpitField="ticket.priority"
direction="OUT">
            <valueMap externalValue="2" cockpitValue="High"/>
            <valueMap externalValue="2" cockpitValue="Medium"/>
            <valueMap externalValue="2" cockpitValue="Low"/>
```

```
        </lookup>
    </mappers>
</create>
```

- This part corresponds to the mapping of Cockpit ITSM incident creations => ServiceNow, with the principle being the same as in the previous section.
- All "direction" attributes have the value "**OUT**".
- The ticket creation time (from the "ticket.creationDate" field) is sent as a timestamp. The ServiceNow user in charge of creating the ticket will convert the timestamp to the correct format.
- The Cockpit ITSM priorities are associated with the ServiceNow urgencies. But since impacts do not exist in Cockpit ITSM, you must send them yourself. In this example, we will send an impact of "2" regardless of the priority of the Cockpit ITSM ticket. The configuration must be adapted in accordance with the particular context.
- End of the ticket creation process.

D. Updating a ticket

```
<update discriminatorField="__operation__" discriminatorValues="UPDATE" attachmentPrivacy="PUBLIC" direction="BOTH">
```

- The <update /> tag is used to update tickets.
- "discriminatorField" **attribute:** the "__operation__" value indicates that we are interested in operations.
- "discriminatorValue" **attribute**: the "UPDATE" value indicates that we will only take update operations into consideration.
- "direction" attribute: the updating of the tickets will be done in both directions, from ServiceNow to Cockpit ITSM and vice versa.

```
    <filters>
        <filter direction="IN">
            <filterRule field="sys_updated_by" value="cockpit" inverted="true"/>
        </filter>
        <filter direction="OUT">
            <filterRule field="messageStatus" value="PUBLIC"/>
        </filter>
    </filters>
```

- Just like for the creation of incidents, a filter is applied to the incoming tickets to disregard the ServiceNow tickets created by the "cockpit" user and thus avoid a loop.
- A filter is applied to the outgoing tickets to only take into account those Cockpit ITSM ticket upgrades which are performed with a confidentiality level of "Public".

1. Mapping of ServiceNow statuses and priorities => Cockpit

```
<mappers>
        <copy externalField="comments" cockpitField="message" stripHtml="true" direction="IN"/>
```

```
            <copy externalField="sys_updated_on" cockpitField="date" direction="IN"/>
            <lookup targetType="status" externalField="state" cockpitField="newStatus" direction="IN">
               <valueMap externalValue="1" cockpitValue="OP_NEW"/>
               <valueMap externalValue="2" cockpitValue="OP_PROCESS"/>
               <valueMap externalValue="3" cockpitValue="OP_WAITING"/>
               <valueMap externalValue="6" cockpitValue="CL_TEMP"/>
               <valueMap externalValue="7" cockpitValue="CL_DEF"/>
               <valueMap externalValue="8" cockpitValue="CL_REJECTED"/>
               <valueMap externalValue="*" cockpitValue="OP_NEW"/>
            </lookup>
            <lookup targetType="priority" externalField="priority" cockpitField="ticket.priority" direction="IN">
               <valueMap externalValue="1" cockpitValue="High"/>
               <valueMap externalValue="2" cockpitValue="High"/>
               <valueMap externalValue="3" cockpitValue="Medium"/>
               <valueMap externalValue="4" cockpitValue="Low"/>
            </lookup>
```

- Same operation as for the creation of tickets.

2. Mapping of Cockpit statuses and priorities => ServiceNow

```
            <set externalField="__type__" cockpitField="N/A" value="INCIDENT" direction="OUT"/>
            <copy externalField="number" cockpitField="ticket.externalReference" direction="OUT"/>
            <copy externalField="comments" cockpitField="message" stripHtml="true" direction="OUT"/>
            <lookup targetType="status" externalField="state" cockpitField="ticket.status" direction="OUT">
               <valueMap cockpitValue="OP_NEW" externalValue="1"/>
               <valueMap cockpitValue="OP_TODO" externalValue="1"/>
               <valueMap cockpitValue="OP_PROCESS" externalValue="2"/>
               <valueMap cockpitValue="OP_WAITING" externalValue="3"/>
               <valueMap cockpitValue="CL_TEMP" externalValue="6"/>
               <valueMap cockpitValue="CL_DEF" externalValue="7"/>
               <valueMap cockpitValue="CL_REJECTED" externalValue="8"/>
               <valueMap cockpitValue="*" externalValue="1"/>
            </lookup>

            <lookup targetType="priority" externalField="urgency" cockpitField="ticket.priority"
direction="OUT">
               <valueMap externalValue="1" cockpitValue="High"/>
               <valueMap externalValue="2" cockpitValue="Medium"/>
               <valueMap externalValue="3" cockpitValue="Low"/>
            </lookup>

            <lookup targetType="priority" externalField="impact" cockpitField="ticket.priority"
direction="OUT">
               <valueMap externalValue="2" cockpitValue="High"/>
               <valueMap externalValue="2" cockpitValue="Medium"/>
               <valueMap externalValue="2" cockpitValue="Low"/>
            </lookup>
         </mappers>
      </update>
```

- Same operation as for the creation of tickets.

E. Closing tickets

```
<close discriminatorField="state" discriminatorValues="6,7" attachmentPrivacy="PUBLIC"
direction="BOTH">
    <filters>
      <filter direction="IN">
        <filterRule field="sys_updated_by" value="cockpit" inverted="true"/>
      </filter>
    </filters>
```

- The <close /> tag is used to close tickets.
- The ServiceNow ticket statuses "6" and "7" correspond to the closing of a ticket.
- Ticket closures can be performed from both Cockpit ITSM and ServiceNow ("BOTH" parameter).
- A filter lets you disregard those updates which are performed by the ServiceNow "cockpit" user.

1. Mapping of ServiceNow fields => Cockpit ITSM

```
    <mappers>
      <copy externalField="close_notes" cockpitField="ticket.response" stripHtml="true" direction="IN"/
>
      <copy externalField="closed_at" cockpitField="date" direction="IN"/>
      <copy externalField="resolved_at" cockpitField="date" direction="IN"/>
      <lookup targetType="status" externalField="state" cockpitField="newStatus" direction="IN">
        <valueMap externalValue="6" cockpitValue="CL_TEMP"/>
        <valueMap externalValue="*" cockpitValue="CL_DEF"/>
      </lookup>
      <map externalField="state" cockpitField="type" direction="IN">
        <valueMap externalValue="6" cockpitValue="CLOSE_TEMP"/>
        <valueMap externalValue="7" cockpitValue="CLOSE_DEF"/>
      </map>
```

2. Mapping of Cockpit ITSM fields => ServiceNow

```
      <set externalField="__type__" cockpitField="N/A" value="INCIDENT" direction="OUT"/>
      <copy externalField="number" cockpitField="ticket.externalReference" direction="OUT"/>
      <copy externalField="comments" cockpitField="ticket.response" stripHtml="true"
direction="OUT"/>
      <copy externalField="close_notes" cockpitField="ticket.response" stripHtml="true"
direction="OUT"/>
      <copy externalField="closed_at" cockpitField="date" direction="OUT"/>
      <set externalField="active" cockpitField="N/A" value="false" direction="OUT"/>
      <lookup targetType="status" externalField="state" cockpitField="ticket.status" direction="OUT">
        <valueMap externalValue="7" cockpitValue="CL_TEMP"/>
        <valueMap externalValue="7" cockpitValue="CL_DEF"/>
        <valueMap externalValue="8" cockpitValue="CL_REJECTED"/>
      </lookup>
      <lookup targetType="status" externalField="incident_state" cockpitField="ticket.status"
direction="OUT">
        <valueMap externalValue="7" cockpitValue="CL_TEMP"/>
```

```
            <valueMap externalValue="7" cockpitValue="CL_DEF"/>
            <valueMap externalValue="8" cockpitValue="CL_REJECTED"/>
          </lookup>

          <lookup targetType="priority" externalField="urgency" cockpitField="ticket.priority"
direction="OUT">
            <valueMap externalValue="1" cockpitValue="High"/>
            <valueMap externalValue="2" cockpitValue="Medium"/>
            <valueMap externalValue="3" cockpitValue="Low"/>
          </lookup>

          <lookup targetType="priority" externalField="impact" cockpitField="ticket.priority"
direction="OUT">
            <valueMap externalValue="2" cockpitValue="High"/>
            <valueMap externalValue="2" cockpitValue="Medium"/>
            <valueMap externalValue="2" cockpitValue="Low"/>
          </lookup>
        </mappers>
      </close>
    </process>
```

IV. Synchronization process - Request

As the synchronization process for requests is similar to that of incidents, it is not described here.

Please see the XML file for an example of a request ticket synchronization.

V. Synchronization process - Change

- In the following example, we will describe the process for synchronizing "Change" tickets.

A. Ticket type

```
<process ticketType="CHANGE" discriminatorField="__type__" discriminatorValue="CHANGE">
```

- Beginning of the process for synchronizing change requests.

```
<ticketIdMap externalField="number" cockpitField="ticket.externalReference"/>
```

- As is the case during the creation of a ticket, first we verify that the operation involves a new ticket.

B. Creating a ticket

```
      <create direction="IN" attachmentPrivacy="ALL">
        <filters>
          <filter direction="IN">
            <filterRule field="sys_created_by" value="cockpit" inverted="true"/>
            <filterRule field="state" value="-2"/>
          </filter>
        </filters>
```

- Incoming tickets do not result in the creation of a Cockpit ITSM ticket unless the ServiceNow status is set to "Scheduled" ("-2").
- A filter lets you disregard those updates which are performed by the ServiceNow "cockpit" user.

1. Mapping of ServiceNow fields => Cockpit

```
        <mappers>
          <copy externalField="number" cockpitField="ticket.externalReference" direction="IN"/>
          <lookup targetType="status" externalField="state" cockpitField="ticket.status" direction="IN">
              <valueMap externalValue="*" cockpitValue="OP_NEW"/>
          </lookup>
          <copy externalField="short_description" cockpitField="ticket.title" stripHtml="true" direction="IN"/>
          <template externalField="" cockpitField="ticket.request" direction="IN"><!
[CDATA[$data["description"]
Justification
=============
$data["justification"]

Risk and impact analysis
========================

$data["risk_impact_analysis"]

Implementation plan
===================
$data["implementation_plan"]

Test plan
=========
$data["test_plan"]

Backout plan
============
$data["backout_plan"]]]></template>
          <copy externalField="opened_at" cockpitField="ticket.creationDate" direction="IN"/>
          <lookup targetType="priority" externalField="priority" cockpitField="ticket.priority" direction="IN">
            <!-- Request items are not prioritized in ServiceNow -->
            <valueMap externalValue="*" cockpitValue="Medium"/>
          </lookup>

          <lookup targetType="team" externalField="assignedTeam" cockpitField="ticket.assignedTeam"
direction="IN">
            <valueMap externalValue="*" cockpitValue="DEVELOPMENT"/>
          </lookup>
        </mappers>
      </create>
```

- Change requests are not prioritized in ServiceNow; all priorities are set to "Medium" in Cockpit ITSM.

- The various phases of a ServiceNow change request (justification, implementation plan, etc.) are injected into the "Message" field of the Cockpit ITSM ticket. This field corresponds to the JavaScript code in the <template /> tag.
- The tickets are assigned to the "DEVELOPMENT" team.

C. Updating a ticket

```
    <update discriminatorField="operation" discriminatorValues="UPDATE" attachmentPrivacy="ALL"
direction="BOTH">
        <filters>
          <filter direction="IN">
            <filterRule field="sys_updated_by" value="cockpit" inverted="true"/>
          </filter>
        </filters>
```

- A filter lets you disregard those updates which are performed by the ServiceNow "cockpit" user.

1. The mapping of fields

```
        <mappers>
          <!--                                  -->
          <!-- ServiceNow => Cockpit   -->
          <!--                                  -->
          <copy externalField="comments" cockpitField="message" stripHtml="true" direction="IN"/>
          <copy externalField="sys_updated_on" cockpitField="date" direction="IN"/>
          <lookup targetType="status" externalField="state" cockpitField="newStatus" direction="IN">
            <valueMap externalValue="-2" cockpitValue="OP_TODO"/>
            <valueMap externalValue="-1" cockpitValue="OP_PROCESS"/>
            <valueMap externalValue="0" cockpitValue="Review"/>
          </lookup>

          <!--                                   -->
          <!-- Cockpit => ServiceNow   -->
          <!--                                   -->
          <set externalField="__type__" cockpitField="N/A" value="CHANGE" direction="OUT"/>
          <copy externalField="number" cockpitField="ticket.externalReference" direction="OUT"/>
          <copy externalField="comments" cockpitField="message" stripHtml="true" direction="OUT"/>
          <lookup targetType="status" externalField="state" cockpitField="ticket.status" direction="OUT">
            <valueMap externalValue="-2" cockpitValue="OP_TODO"/>
            <valueMap externalValue="-1" cockpitValue="OP_PROCESS"/>
            <valueMap externalValue="0" cockpitValue="CL_TEMP"/>
            <valueMap externalValue="0" cockpitValue="CL_DEF"/>
            <valueMap externalValue="4" cockpitValue="CL_REJECTED"/>
          </lookup>
        </mappers>
      </update>
```

D. Closing of a ticket

1. Filters

```
<close discriminatorField="state" discriminatorValues="3,4" direction="BOTH">
  <filters>
    <filter direction="IN">
      <filterRule field="sys_updated_by" value="cockpit" inverted="true"/>
    </filter>
  </filters>
```

- Only the ServiceNow statuses "3" and "4" (Closed and Canceled) will trigger the closing of a ticket.
- A filter lets you disregard those updates which are performed by the ServiceNow "cockpit" user.

2. The mapping of fields

```
  <mappers>
    <!--                                -->
    <!-- ServiceNow => Cockpit   -->
    <!--                                -->
    <copy externalField="comments" cockpitField="message" stripHtml="true" direction="IN"/>
    <copy externalField="close_notes" cockpitField="ticket.response" stripHtml="true" direction="IN"/
>
    <copy externalField="closed_at" cockpitField="date" direction="IN"/>
    <lookup targetType="status" externalField="state" cockpitField="newStatus" direction="IN">
      <valueMap externalValue="*" cockpitValue="CL_DEF"/>
    </lookup>

    <!--                                -->
    <!-- Cockpit => ServiceNow   -->
    <!--                                -->
    <set externalField="__type__" cockpitField="N/A" value="CHANGE" direction="OUT"/>
    <copy externalField="number" cockpitField="ticket.externalReference" direction="OUT"/>
    <copy externalField="comments" cockpitField="ticket.response" stripHtml="true"
direction="OUT"/>
    <lookup targetType="status" externalField="state" cockpitField="ticket.status" direction="OUT">
      <valueMap externalValue="0" cockpitValue="CL_TEMP"/>
      <valueMap externalValue="0" cockpitValue="CL_DEF"/>
      <valueMap externalValue="4" cockpitValue="CL_REJECTED"/>
    </lookup>
  </mappers>
 </close>
</process>
</ticketSync>
```

- The </ticketSync> tag marks the end of the synchronization.

End of document