



cockpit
IT Service Manager

Ticketing - Synchronization Guide

FAQ document

Table of contents

Introduction.....	3
I.Objective.....	3
II.Prerequisite.....	3
III.Functioning.....	3
IV.Types of connectors.....	3
Configuring the Cockpit ITSM portal.....	4
I.Configuration.....	4
II.Utilization.....	4
III.Logs.....	5
Configuring the XML.....	7
I.Structure.....	7
A.General configuration.....	7
B.Configuration of the connector.....	7
C.Process configuration.....	8
II.Configuration of operations.....	9
A.Configuration of the operation.....	10
B.Configuration of filters.....	10
C.Configuration of mappers.....	11
Annex.....	16
I.Detecting operation types.....	16
II.Cockpit ITSM fields.....	16

Introduction

I. Objective

- Acquaint users with the "Synchronization" menu of the "Ticketing" module.

II. Prerequisite

To understand how the synchronization of the Tickets module works, it is recommended that you:

- have a firm understanding of the Tickets module in Cockpit ITSM (priorities, environments, statuses, etc.);
- understand basic XML (Extensible Markup Language);
- master all the functions of the external system to be synchronized with Cockpit ITSM.

III. Functioning

All communication between the Cockpit ITSM portal and the external system takes place via a connector and an XML file:

- Entry (External system => Cockpit ITSM): The connector retrieves the ticket data from the external system and transforms this data into a list of keys/values inside an XML file. Cockpit ITSM then retrieves this data and incorporates it into its tickets.
- Exit (Cockpit ITSM => External system): Cockpit sends a list of key/value combinations to the connector in an XML file, and transforms this list into corresponding operations in the external system.
- Each connector can only be used with one external system. To synchronize Cockpit ITSM with an external system that does not yet have a connector, you must first open a support ticket to request a new connector.

IV. Types of connectors

List of external systems that have connectors:

- ServiceNow (ID: "service_now")
- Jira Software (ID: "jira")

Configuring the Cockpit ITSM portal

Menu: Tickets > Module configuration > Synchronization

Objective: Manage the external systems which are to be synchronized with the tickets of the Cockpit ITSM portal.

I. Configuration

Properties of a synchronization with an external system:

Field	Description
Organization	A synchronization is associated with an organization. There can only be one synchronization per organization. Mandatory
Application	Free text field for specifying the name of the external system that is synchronized with the Cockpit ITSM portal, e.g. ServiceNow Mandatory
End-user	Specify an active Cockpit ITSM end-user belonging to the organization. Tickets synchronized with the external system will be created in Cockpit ITSM with this end-user. Mandatory
Status	Active/Inactive When the status is inactive, the tickets are not synchronized. When a synchronization is deactivated and then reactivated, tickets created or modified during the period of inactivity are not taken into account. It is possible to modify this process and take into account actions that occurred during the period of inactivity. If this need exists, contact the ITSM Cockpit Support.
Configuration data (XML)	Enter the source code of the XML file directly into this field. This code will be used by the connector to communicate with the Cockpit ITSM portal and the external system. Mandatory

II. Utilization

Actions:

- Reload: Refresh the display the dates and times of the last executions in success and failures.
- Delete a synchronization.
- Copy a synchronization to create a new one.

The "Last execution" field indicates the date and time of the last synchronization between the 2 applications.

The "Last failure" field indicates the date and time of the last synchronization that could not be performed. If the date of the last execution is more recent than the date of the last failure, the synchronization is technically operational.

III. Logs

Important: Two cases arise:

- The log directory is located on the machine hosting the portal, if you are in SaaS mode you do not access the synchronization logs.
- If you are in Premium mode, you can specify in the XML file the path where you want to store the log file.

Below is a non-exhaustive list of errors that may occur in the logs.

1. Connection error

HttpClientErrorException.Unauthorized: 401 Unauthorized

The Cockpit portal could not connect to the target application, the connection is considered failed in the synchronization management menu and will be noted in the "Last failure" field.

It is necessary to check everything related to identification and access:

- Application access Url
- Roles, ID and password of the target application
- etc.

2. Synchronization error

[TicketSyncJob] Synchronisation stats :

Operations received: 1

Incoming operations successfully synchronized: 0

Outgoing operations successfully synchronized: 0

Incoming operations failed to synchronize: 0

Outgoing operations failed to synchronize: 1

The "Operations received" field indicates the number of actions to be synchronized between the 2 systems.

The following terms indicate the meaning of the elements to be synchronized:

- "Incoming... ": From the external system to Cockpit ITSM
- "Outgoing.... ": From Cockpit ITSM to the external system

In this example there was a synchronization operation to be performed from Cockpit ITSM to the external system and it failed.

In the menu, the synchronization does not appear in the failure field but in "Last execution", because technically the synchronization worked, if elements were not synchronized, it is due to the system settings or to the XML file.

Details of incoming and outgoing operations also appear in the log, below are some examples:

Unexpected exception while handling ticket log com.cockpit.model.ticket.ticketElements.TicketLog@6f5

An element could not be synchronized, it is necessary to check the configuration, especially if all the fields associated with each other in the XML file exist in their respective systems.

[updateTicket] Incoming ticket does not match filters => Skipping ticket line INCXXXXXXXX

In this example an element was not taken into account during synchronization because it does not match the filters set in the XML file.

Configuring the XML

Objective: Introduce the XML file structure that is universal for all connectors. Please see the connector FAQs for information on the special features of each particular connector.

I. Structure

In general, the configuration of the XML file is structured into 3 parts.

- The general configuration: <ticketSync> tag
- The configuration of the connector: <connector> tag
- The configuration of the process: <process> tag

A. General configuration

The <ticketSync> tag contains the following elements:

Parameter	Description
xmlns	XML Namespace (xmlns): Do not modify this value. The "xmlns" makes it possible to verify the presence of a conflict in the XML code which you have entered into the "Configuration data (XML)" field. If the XML is configured incorrectly, an error message will appear while saving the data. Example: "Invalid XML configuration: Invalid content was found starting with element 'control'."
description	Text field for specifying the name of the external system. Example: "Service Now Jakarta"
SyncFrequency-InMinutes	Indicates the frequency of the synchronization in minutes.
lastUpdateField	Complete the field (a timestamp) to allow for the latest operation of each mapping to be identified. This field lets you take operations into account in the correct chronological order during synchronizations.

Example:

```
<ticketSync xmlns="http://www.cockpit-itsm.com/TicketSyncConfiguration"
description="Système externe" syncFrequencyInMinutes="5">
<connector ... />
<lastUpdateField>Nom_du_Champ</lastUpdateField>
<process ... />
</ticketSync>
```

B. Configuration of the connector

The <connector> tag contains the following elements:

Parameter	Description	Default value
id	ID corresponding to the type of connector. This information is provided in the introduction to this document.	
username	User of the external system that the connector uses to connect to the external system.	
password	The user's password (previous field).	
url	The external system URL used by the connector.	
defaultTimeZone	Default time zone used to interpret the date fields of the external system; does not take into account the actual time zone of these fields. For the sake of simplification, you can set everything to UTC (fields of the external system and the Cockpit ITSM end-user used to create tickets).	UTC
dateFormat	Date/hour format used for exchanges between timestamps.	

Example:

```
<connector id="service_now" username="cockpit" password="secret"
url="https://xxxxxx.service-now.com" defaultTimeZone="UTC" dateFormat="yyyy-MM-dd
HH:mm:ss">
  <parameters>
    <entry>
      <string>traceDirectory</string>
      <string>/Users/Username/Temp/Service Now</string>
    </entry>
  </parameters>
</connector>
```

In this example, the configuration corresponds to a "ServiceNow" connector.

Note: The "traceDirectory" parameter makes it possible to specify a path for saving synchronization logs. This path is located on the machine that hosts the portal, which is not accessible in SaaS mode. Consequently, this parameter is only useful when you are in "on-premise" mode.

C. Process configuration

"Process" is used in the ITIL sense of the word. Currently, 3 types of processes are available:

- INCIDENT
- REQUEST
- CHANGE

These are the 3 types of tickets you can create in the Cockpit ITSM portal. Depending on the configuration, you can therefore choose to synchronize one or more types of tickets.

The process configuration must be described in the "**<process />**" tag.

Process parameters:

Parameter	Description
ticketType	The type of ticket you wish to synchronize: <ul style="list-style-type: none"> • INCIDENT • REQUEST • CHANGE
discriminatorField	Attribute making it possible to identify the field that will be used to find the process entered in "ticketType". Example: "__type__" corresponding to the type of ticket.
discriminatorValue	Value used to identify the process. Example: "CHANGE" for "change" tickets.
TicketIdMap parameters	
externalField	Field that makes it possible to identify the ID of the ticket in the mapping.
cockpitField	Field that makes it possible to identify the ID of the ticket in the Cockpit ITSM portal.

Example:

```
<process ticketType="CHANGE" discriminatorField="__type__" discriminatorValue="CHANGE">
  <ticketIdMap externalField="XXXX" cockpitField="ticket.externalReference"/>
  <create ... />
  <update ... />
  <close ... />
</process>
```

In this example, the "CHANGE" process is used for all tickets having a "__type__" value of "CHANGE".

The <ticketIdMap> tag lets you combine all the IDs of a ticket:

- The "externalField" attribute contains the ID of the ticket in the external system.
- The "cockpitField" attribute contains "ticket.externalReference," which contains the ID of the ticket in Cockpit ITSM.

Important: We usually use the notation "__KEY__" for the keys that are injected into the external system by the connector. This makes it possible to distinguish them from the KEY values originating from the external system. Nevertheless, these 2 notations refer to the same value.

II. Configuration of operations

Principles:

The following operations may be performed on tickets:

- CREATE (creation of a ticket)

- UPDATE (adding information to a ticket)
- CLOSE (closure of a ticket)

The configuration of an operation consists of:

- the configuration of the operation;
- filters: used to filter out the tickets which you do not wish to be taken into account during the operation;
- mappers: used to transform the ticket data into a tag/attribute description that can be understood by the connector.

A. Configuration of the operation

Details:

Parameters of CREATE/UPDATE/CLOSE actions	
Parameter	Description
direction	Direction in which the action is executed: <ul style="list-style-type: none"> • IN (External system => Cockpit ITSM) • OUT (Cockpit ITSM => External system) • BOTH
attachmentPrivacy	Lets you filter outgoing attachments according to their access rights: <ul style="list-style-type: none"> • ALL • PUBLIC • NONE

Example:

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

B. Configuration of filters

How filters work:

- Filters are applied one by one in the order of their appearance in the XML code.
- Tickets are only synchronized if the filter conditions are met.
- A filter is only applied to tickets in the specified direction:
 - IN: for incoming tickets
 - OUT: for outgoing tickets
 - BOTH: this value is not valid for filter tags
- Filter conditions can be reversed.

Details:

<filter> tag

Attribute	Description
direction	If an action already has a "direction" parameter, then it is only possible to add a filter to one direction: <ul style="list-style-type: none"> • IN (External system => Cockpit ITSM) • OUT (Cockpit ITSM => External system) • BOTH (Do not use for filters)
<filterRule> tag	
Attributes	Description
field	Name of the field that was searched
value	Value of the field that was searched
inverted	Reverses the value that was found

Example:

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

- This filter is applied to all incoming tickets (External system => Cockpit ITSM).
- The filter takes the ticket creation operations into account.
- The filter searches for tickets created by the "cockpit" user and excludes those with an "inverted" attribute. It therefore takes into account all the tickets which have not been created by the user "cockpit".

C. Configuration of mappers

Principles:

- A list of "mappers" is configured for each operation to map the Cockpit ITSM fields with the fields of the external system (and vice versa). Mappers describe the actions which will be performed by the operation.
- Mappers are applied one by one in the order of their appearance in the XML file. They are used to:
 - modify the data so it can be sent to the external system (outgoing operations);
 - modify Cockpit ITSM tickets to incorporate the data originating from the external system (incoming operations).

Details:

Available mappers	
Name	Description
set	Assigns a fixed value to a field

copy	Copies a field without modifying its value
lookup	Uses a mapping table to link a field from the external system to a field from Cockpit ITSM (and vice versa). Currently, the following ticket fields may be synchronized: <ul style="list-style-type: none"> • Status • Priority • Application • Environment • Team
user	Links a user of the external system to a user of Cockpit ITSM
map	Links one field to another according to a static mapping table
template	Uses a Velocity template to link one field to another. The data is placed in the "\$data" variable.

Example:

```
<mitters>
  <copy externalField="number" cockpitField="ticket.externalReference" direction="IN"/>
  <set externalField="__type__" cockpitField="N/A" value="INCIDENT" direction="OUT"/>
</mitters>
</create>
```

- For incoming operations, the externalField "number" of the external system is linked to the cockpitField "ticket.externalReference" (these are the ticket IDs).
- For outgoing operations, the externalField "__TYPE__" is set to "INCIDENT" (the external system will therefore create an INCIDENT ticket).

Detailed list of mappers:

1. "set" mapper

Configuration of the "set" mapper	
Name	Description
externalField	Field of the external system
cockpitField	Cockpit ITSM field
value	Value to be assigned to the field; this value is static.
direction	Direction of the mapper: <ul style="list-style-type: none"> • IN (External system => Cockpit ITSM) • OUT (Cockpit ITSM => External system) • BOTH

2. "copy" mapper

Configuration of the "copy" mapper	
Name	Description
externalField	Field of the external system
cockpitField	Cockpit ITSM field
direction	Direction of the mapper: <ul style="list-style-type: none"> • IN (External system => Cockpit ITSM) • OUT (Cockpit ITSM => External system) • BOTH

3. "lookup" mapper

Configuration of the "lookup" mapper	
Name	Description
externalField	Field of the external system
cockpitField	Cockpit ITSM field
direction	Direction of the mapper: <ul style="list-style-type: none"> • IN (External system => Cockpit ITSM) • OUT (Cockpit ITSM => External system) • BOTH
targetType	Type of field: <ul style="list-style-type: none"> • status • priority • application • environment • team
valueMap	Static mapping table; the (*) symbol may be used to accept all types of data. The values of the fields listed in targetType (see above) are those which can be mapped.
valueMap[].externalValue	External system value
valueMap[].cockpitValue	Corresponding Cockpit ITSM value

4. "user" mapper

Configuration of the "user" mapper	
Name	Description
externalField	Field of the external system
cockpitField	Cockpit ITSM field

direction	Direction of the mapper: <ul style="list-style-type: none"> • IN (External system => Cockpit ITSM) • OUT (Cockpit ITSM => External system) • BOTH
property	Property of the Cockpit ITSM user to be compared (login, fullName, externalReference). For incoming data, only the "login" must be unique.
userType	Possible user types: <ul style="list-style-type: none"> • CONTACT (End-user) • OPERATOR

5. "map" mapper

The "map" mapper is similar to the "lookup" mapper, except the "lookup" mapper allows for the mapping of Cockpit ITSM values (team, status, etc.).

"map" makes it possible to map other values such as character strings, numbers, dates, etc.

Configuration of the "map" mapper	
Name	Description
externalField	Field of the external system
cockpitField	Cockpit ITSM field
direction	Direction of the mapper: <ul style="list-style-type: none"> • IN (External system => Cockpit ITSM) • OUT (Cockpit ITSM => External system) • BOTH
valueMap	Static mapping table; the (*) symbol may be used to accept all types of data. The values of the fields listed in targetType (see above) are those which can be mapped.
valueMap[].externalValue	External system value
valueMap[].cockpitValue	Corresponding Cockpit ITSM value

6. "template" mapper

Principles:

- The "template" mapper uses Velocity Template Language to inject data into a text that is predetermined by a variable.
Example: Retrieving the ticket title or ID to inject it into the title field of the ticket in the other system.
- Reference document for VTL (Velocity Template Language):
<https://velocity.apache.org/engine/2.0/user-guide.html>
- It is also possible to add plain text.

Details:

Configuration of the "map" mapper	
Name	Description
externalField	Field of the external system
cockpitField	Cockpit ITSM field
direction	Direction of the mapper: <ul style="list-style-type: none"> • IN (External system => Cockpit ITSM) • OUT (Cockpit ITSM => External system) • BOTH
<element content>	Indicates a Velocity template; the template must be integrated into a CDATA section.

Example:

```

<![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"]
]]>

```

Annex

I. Detecting operation types

The following steps will help you determine whether an incoming operation is a creation, an update, or a closure:

- Has the ID of the incoming ticket already been mapped with a Cockpit ID?
- If no mapping exists, then a ticket is created.
- If the ticket already exists, check to see if the operation is a closure.
- If the operation is indeed a closure, then the ticket is closed.
- In all other cases, the operation is an update.

II. Cockpit ITSM fields

The following are the most frequently used fields:

Name	Type	Description
ticket.externalReference	String	External reference of a ticket (example: ID of a ServiceNow ticket)
ticket.status	Lookup	Ticket status
ticket.title	String	Ticket title
ticket.request	String	Ticket request field (creation)
ticket.response	String	Ticket response field (closure)
ticket.creationDate	Timestamp	Ticket creation date
ticket.priority	Lookup	Ticket priority
ticket.assignedTeam	Lookup	Team in charge of tickets
message	String	Messages exchanged in the tickets
date	Timestamp	Dates of the exchanges in the tickets
newStatus	Lookup	New ticket status
type	Enum	The various types of exchanges: <ul style="list-style-type: none"> • ACKNO • ASK_VALID • ASSIGN • CLOSE_DEF • CLOSE_TEMP • CREATION • EMAIL_RECEIVED • EMAIL_SENT • MESSAGE • MIGRATION • REJECT • REOPEN • SAVE • TRANSFER

		<ul style="list-style-type: none">• UNASSIGN• UPDATE_AUTHOR• VALIDATE
--	--	---

End of document