

Oracle® Enterprise Manager Jira Service Management Connector Installation and Configuration Guide



24ai Release 1 (24.1)

F97232-01

December 2024

The Oracle logo, consisting of the word "ORACLE" in white, uppercase, sans-serif font, centered within a solid red square.

ORACLE®

Oracle Enterprise Manager Jira Service Management Connector Installation and Configuration Guide, 24ai Release 1
(24.1)

F97232-01

Copyright © 2022, 2024, Oracle and/or its affiliates.

Contents

Preface

Audience	v
Documentation Accessibility	v
Related Resources	v
Conventions	vi

1 Introduction to the Jira Connector

Versions Supported	1-1
Ticket Generation Contents	1-1
Auto Ticketing	1-2
Manual Ticketing	1-2
Ticket Templates	1-2
Grace Period	1-3
Retry	1-3

2 Installing and Configuring the Jira Connector

Prerequisites	2-1
Installing the Connector	2-2
Creating a Connector Instance	2-2
Configuring the Connector	2-3
Enabling or Disabling the Connector	2-4
Providing General Settings	2-4
Working with Ticket Templates	2-5
Adding Ticket Templates	2-6
Viewing Template Code	2-7
Removing a Template	2-7
Editing Templates	2-7
Restoring Templates	2-8
Deleting the Connector	2-8

3 Creating Jira Tickets

Automatically Creating a Ticket	3-1
Manually Creating a Ticket	3-4

4 Using Default Templates

Template Process	4-1
Reading Ticket Templates	4-1
Customizing Ticket Templates	4-9
Defining New Templates	4-10

Index

Preface

This *Connector Configuration* guide provides the information that you require to configure Management Connectors that integrate Enterprise Manager with other management tools and help desk systems.

Audience

This guide is written for Oracle Database system administrators who want to configure Management Connectors to enable integration between Enterprise Manager and other systems.

You should already be familiar with Oracle Enterprise Manager.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Resources

For more information, see the following books in the Oracle Enterprise Manager documentation set:

- *Oracle Enterprise Manager Cloud Control Basic Installation Guide*
- *Oracle Enterprise Manager Cloud Control Advanced Installation and Configuration Guide*
- *Oracle Enterprise Manager Cloud Control Administrator's Guide*
- *Oracle Enterprise Manager Cloud Control Upgrade Guide*
- *Oracle Enterprise Manager Framework, Host, and Services Metric Reference Manual*
- *Oracle Enterprise Manager Command Line Interface*
- *Oracle Enterprise Manager Cloud Control Extensibility Programmer's Guide*
- *Oracle Database 2 Day DBA*

The latest versions of this and other Oracle Enterprise Manager documentation can be found at:

<http://docs.oracle.com/en/enterprise-manager/>

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at:

<http://docs.oracle.com/en/enterprise-manager/<13.5>>

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction to the Jira Connector

The Oracle Management Connector for Jira Management Service REST Ticketing Connector integrates Jira with Enterprise Manager through a REST connection. Using this connector, you can create and update a ticket for any incident created in Enterprise Manager.

The ticket generated by the connector contains the relevant information about the Enterprise Manager incident, including a link to the Enterprise Manager console to enable helpdesk analysts to leverage Enterprise Manager's diagnostic and resolution features to resolve the incident. In Enterprise Manager, the ticket ID and its link to navigate to the Jira Ticket is shown in the context of the incident. This easily enables Enterprise Manager administrators to quickly access the ticket.



Note:

Before proceeding, see the Using Incident Management chapter in the *Oracle Enterprise Manager Cloud Control Administrator's Guide* to understand the event and incident management concepts.

The following sections explain various Jira Connector concepts that you must understand before you start using the Jira Connector.

- [Versions Supported](#)
- [Ticket Generation Contents](#)
- [Auto Ticketing](#)
- [Manual Ticketing](#)
- [Ticket Templates](#)
- [Grace Period](#)
- [Retry](#)

Versions Supported

The Jira Connector communicates through version 3 of the REST API component of Jira and functions with Oracle Enterprise Manager Cloud Control 13c Release 5 (13.5.0.0.0) and later versions.

Ticket Generation Contents

The ticket generated by the connector contains the relevant information about the Enterprise Manager incident, including a link to the Enterprise Manager console to enable help desk analysts to leverage Enterprise Manager's diagnostic and resolution features to resolve the incident.

In Enterprise Manger, the ticket ID, and link to the Jira ticket is shown in the context of the incident. This provides Enterprise Manager administrators an easy way to quickly access the ticket.

Auto Ticketing

Auto ticketing refers to creating or updating tickets automatically for any matching rule(s) in Enterprise Manager. You can define a set of event or incident rules for which tickets must be opened/updated for changes in underlying event or incident attributes. For example, changing event severity from Warning to Critical can update the associated ticket.



See Also:

Using Incident Management chapter in the *Oracle Enterprise Manager Cloud Control Administrator's Guide*.

After the ticket is opened, any subsequent update of the incident attributes or underlying event attributes, such as a change in underlying events severity, updates the ticket. After the incident is cleared in Enterprise Manager, the ticket is updated and you can optionally go to Jira to close the ticket.



See Also:

[Automatically Creating a Ticket](#)

Manual Ticketing

From the Enterprise Manager console, you can manually open a Jira ticket based on an open incident in Enterprise Manager. The Jira Connector populates the ticket with details based on the incident and the ticket template selected.



See Also:

[Manually Creating a Ticket](#)

Ticket Templates

Ticket templates are XML transformation style sheets that transform Enterprise Manager incidents to a Jira ticket format before the requests are sent to Jira. A ticket template specifies how an Enterprise Manager incident and its associated event attributes can be mapped to the ticket attributes of Jira.

In Auto Ticketing, while setting up a rule, you select a configured connector and select the ticket template from the template list. The selected ticket template is used when a ticketing request is sent to the Jira. For manual ticketing, you have to select a connector instance and ticket template before submitting a request for ticket creation.

The Jira Connector includes some out-of-box default ticket templates. You can customize default templates according to your functional needs. Oracle recommends that you back up these factory-built templates before customizing them.



See Also:

[Using Default Templates](#)

Grace Period

The grace period enables you to prevent creating a large number of tickets for frequently created incidents due to re-occurrence of the same event.

For example, an event is raised and causes an incident to be created in Enterprise Manager. The rule defined to create the ticket enforces the ticket to be created in Jira. If the grace period is one hour and the event is cleared at 10:00 a.m., this clears the incident and ticket. If the same event reoccurs before 11:00 a.m. and enforces creation of another incident, the grace period functionality stops creation of a new ticket for this and reopens the same ticket instead.

Retry

The "Retry" section, available on the connector configuration page, enables you to specify whether a failed ticketing request needs to be retried within a configurable expiration time.

Enabling the Retry option gives you the option of specifying whether you want to resend a ticketing request if the request fails the first time, and specifying the time period after which you want to abandon the retry. Enterprise Manager retries every 2 minutes until the request is successful or the retry interval expires.

2

Installing and Configuring the Jira Connector

This chapter provides the following information for setting up and configuring the Jira Connector and related tasks:

- [Prerequisites](#)
- [Installing the Connector](#)
- [Creating a Connector Instance](#)
- [Configuring the Connector](#)
- [Enabling or Disabling the Connector](#)
- [Providing General Settings](#)
- [Working with Ticket Templates](#)
- [Deleting the Connector](#)

Prerequisites

The Jira Connector communicates through Jira's REST API. Before using the Jira Connector, ensure that you meet the following prerequisites:

- Jira User exists with all necessary permissions and personnel information.
- My Oracle Support Preferred Credentials are set. To set the credentials, perform the following steps:
 1. From the Enterprise Manager console Setup menu, select **My Oracle Support**, then **Set Credentials**.
 2. Enter the user name and password information, then click **Apply**.
 3. From the Enterprise Manager console Setup menu, select **Extensibility**, then **Self Update**.
 4. From the Self Update page, look for the Jira Service Management REST Ticketing Connector in the table list. If it does not appear, go to the next step.
 5. Click **Check Updates** to schedule a job to collect the metadata for the connector, as shown in following figure.

Type	Available Updates	Downloaded Updates	Applied Updates	Description
Agent Software	0	0	0	1 Agent software has to be installed on hosts for managing the host.
BI Publisher Reports	0	0	0	0 BI Publisher Reports can be run against Enterprise Manager data in a variety of output formats. These reports can also be scheduled and sent via e-mail.
Compliance Content	0	0	0	0 Compliance Content contains Framework, Standard, Rules with support of add and delete on these entities.
Data Masking and Subsetting Templates	0	0	0	0 Templates for Data Masking, Application data masking and Subsetting.
Diagnostic Checks	0	0	0	0 Target side policy checks that identify conditions that may require the attention of target administrators.
Diagnostic Tools	0	0	0	0 Auto-updatable entity for CRACHECK and Diagnostic Assistant(DA).
EM Deployment Prerequisite Resource	0	0	0	0 EM Deployment Prerequisite Checks are the metadata used for checking prerequisites for install, Upgrade, Patching of EM Platform and Plugins.
EM Update Manager	0	0	0	0 EM Update Manager for managing server-side updates.
Enterprise Configuration Search	0	0	0	0 Configuration search definitions can be used at the enterprise level to search the configuration data of managed targets such as hosts and databases.
Help-on-Monitoring	0	0	0	0 Enterprise Manager Always-On Monitoring.
Extensibility Development Kit	0	0	0	1 Extensibility Development Kit (EDK) is a standalone tool designed to help development of Enterprise Manager extensions (for example, Plug-Ins).
Management Connectors	0	0	0	4 Management Connectors are components that integrate different enterprise frameworks into the Enterprise Manager Console.
Metric Extensions	0	0	0	0 Metric Extensions enhance Enterprise Manager's monitoring capability by allowing you to create new metrics to monitor conditions specific to your environment.
Middleware Profiles and Gold Image	0	0	0	0 A collection of Software Components used for provisioning of Oracle Application Server homes.
Monitoring Templates	0	0	0	0 Monitoring Templates can be used to distribute monitoring across your enterprise by applying a subset of monitoring settings to targets.
Open Center to EM Migration Utility	0	0	0	0 Utility to migrate assets managed by Open Center to Enterprise Manager.

 **Note:**

You can also set up Self-Update in offline mode. See *Applying an Update in Offline Mode* in the *Oracle Enterprise Manager Cloud Control Administrator's Guide* for details.

Installing the Connector

This section provides the procedure required to download and install (apply) the connector. The Jira connector is not included out-of-the-box, so you need to download it from the Oracle Enterprise Manager store. All connectors in the Enterprise Manager store are available on the Self Update page for downloading.

To install the connector:

1. From the Enterprise Manager console Setup menu, select **Setup**, then select **Extensibility**, then select **Self Update**.
2. Click on the **Management Connector** link in the Type column.
The Jira connector appears in the list with a status of *Available*.
3. Click on the appropriate Jira connector to select it, and then click **Download**.
The Schedule Download window appears, where you can determine when the download should be performed.
4. Click **Select** to download immediately.
If you want to schedule the download for a later time, specify the date and time when the download should occur, and click **Select**. You will need to return and finish the remainder of this procedure after the scheduled download date and time.
5. If necessary, refresh the screen until the status of the connector changes to Downloaded.
6. Click on the Jira Service Management REST Ticketing Connector to select it, then click **Apply**.
After you respond to the prompt to confirm the operation, a page appears that indicates the request has been submitted.
7. Refresh the screen until the status of the connector changes from *Apply Scheduled* to *Applied*.

Creating a Connector Instance

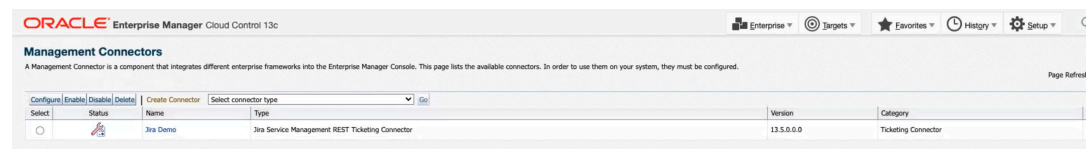
To use the Jira connector, you need to create one or more instances of this connector type. A connector instance refers to the configured instance of the connector with specific details, such as connector name, API Token, Project Key and so forth of the Jira instance. Multiple connector instances for the same connector type are supported, which provides the flexibility to configure multiple Jira instances with Enterprise Manager.

For example, since Enterprise Manager is monitoring production targets and test targets, you can then create two instances of the connector and configure Enterprise Manager so that incidents from production targets create tickets in one connector instance, and incidents from test targets create tickets in the second connector instance.

To create a connector instance:

1. From the Enterprise Manager console, click **Setup**.
2. Select **Extensibility** in the pane.
3. Select **Management Connectors** in the left pane.
4. Select a connector type from the **Create Connector** drop-down list and click the **Go** button.
5. Enter the connector name and description on the next screen and click **OK**.

The connector instance appears as *unconfigured* on the **Management Connectors** page, shown in the following image.



Configuring the Connector

Perform the following steps for basic configuration:

1. As Super Administrator, from the Enterprise Manager console, click **Setup**.
2. Click **Extensibility**, then **Management Connectors**.

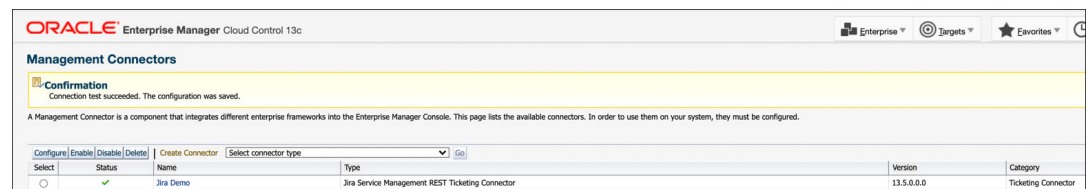
The Management Connectors page appears that lists all created connector instances.

Note:

The status column with a green tick mark indicates that the connector is already configured.

3. Click the **Configure** button for the Jira Connector. The General tab of the Configure Management Connector page appears.
4. Provide the required settings. See [Providing General Settings](#) for details.
5. Click **OK**.

The Management Connectors page reappears. The row for the Jira Connector should have a check mark in the Status column, as shown in the following figure.



6. **Optional:** Register any customized templates using the `emctl register_template connector command` from the `ORACLE_HOME/bin` directory, where `ORACLE_HOME` is the Oracle home directory of the OMS.

You can view all ticket templates by clicking the Ticket Templates tab.

Enabling or Disabling the Connector

You can enable or disable a connector instance without needing to reconfigure or remove the connector from the system.

You can enable or disable a connector from the Management Connector page by selecting the connector and clicking either the **Enable** or **Disable** button. After a connector is disabled, you will see a red cross icon under the Status column. By default, the connector instance is enabled as soon as it is configured.



Note:

Only enabled, connectors are available for ticketing operations.

Providing General Settings

The following sections explain how to provide various configuration details for the General tab of the Configure Management Connector page as shown in the following figure.

The screenshot shows the 'General' tab configuration page. It includes sections for 'Connection Settings' (with fields for createTicket, getTicket, updateTicket, Token, Project Key, Ticket Key), 'Web Console Settings' (with 'Enable web console features' checkbox and 'Jira Subdomain' field), 'Grace Period' (with 'Enable grace period' checkbox and 'Grace Period (Hours)' field), and 'Retry' (with 'Enable Retry' checkbox and 'Expiration (Hours)' field).

Connection Settings

The Jira Connector communicates with the Service Desk through their Web services. Mandatory fields are indicated with an asterisk (*).

- **WebService End Points**—End points to `createTicket`, `updateTicket`, and `getTicket` Web services exposed by Jira. Each of these have a variable called *subdomain* that needs to be updated with your Jira subdomain.
- **Token**—API Token with permissions to retrieve, create, and update tickets from Jira. Note: The API Token must be base64 encoded. See Atlassian’s guide on how to base64 encode

your API Token: <https://developer.atlassian.com/cloud/jira/platform/basic-auth-for-rest-apis/#supply-basic-auth-headers>

On Linux/Unix/MacOS:

```
echo -n user@example.com:api_token_string | base64
```

On Windows (using Powershell):

```
$Text = 'user@example.com:api_token_string'  
$Bytes = [System.Text.Encoding]::UTF8.GetBytes($Text)  
$EncodedText = [Convert]::ToBase64String($Bytes)  
$EncodedText
```

- **Project Key**—Project Key of the project tickets will be created in.
- **Ticket ID**—An existing Ticket ID to test the connection to the Jira system. The ticket does not need to be open.

Web Console Settings

Web Console settings are required if you want the connector to provide links to Jira tickets created by Enterprise Manager in the context of an incident.

To enable this functionality, provide the following Web console settings.

- **Enable Web Console** —Check this box to enable launching of the Jira incident page within context from Enterprise Manager.
- **Jira Subdomain** — The subdomain of your Jira account.

Grace Period

You can enable and disable the grace period and configure its value. By default, the grace period is disabled. See [Grace Period](#) for details. This setting applies to all incidents the Jira Connector processes. The grace period interval accepts the number of hours as any integer value. For example, 1 or 2.

Retry

You can enable and disable the Retry option and configure its interval value. By default, the Retry is disabled. See [Retry](#) for details. This setting applies to all incidents in the Jira Connector processes. The retry period interval accepts the number of hours as any integer value. For example, - 1 or 2.

Working with Ticket Templates

When a connector instance is created in Enterprise Manager, all associated templates are available for ticket create/update operations. There are seven default Jira templates:

- `jiraStoryCreateAndUpdate.xsl`
- `jiraBugCreateAndUpdate.xsl`
- `jiraTaskCreateAndUpdate.xsl`
- `jiraSubtaskCreateAndUpdate.xsl`
- `jiraIncidentCreateAndUpdate.xsl`
- `jiraProblemCreateAndUpdate.xsl`

- JiraServiceRequestCreateAndUpdate.xsl



Tip:

See [Template Process](#) for detailed information about these templates.

The following sections provide information about adding, viewing, removing, and replacing ticket templates.

Adding Ticket Templates

You need to add ticket templates before they are recognized in Enterprise Manager. The connector provides out-of-box templates that are automatically registered when the connector instance is created. If you want to edit these, you can do so directly in Enterprise Manager. However, if you need to use a new template (other than one out-of-box), you need to register the new template using the `emctl` command. You can then edit the template in Enterprise Manager. See [Editing Templates](#) for more information.

To add and register a template:

1. Ensure that you have execute privilege on `emctl` and the ability to read the ticket template.
2. For each template, run the following `emctl register_template` connector command:

```
$ORACLE_HOME/bin/emctl register_template connector
[-t <template.xml>]
[-repos_pwd <repos password>]
[-cname <connectorName>]
[-iname <internalName>]
[-tname <templateName>]
[-ttype <templateType>]
[-d <description>]
```

Where:

```
-t           Template(full path)
-repos_pwd  Enterprise Manager Root (SYSMAN) Password
-cname      Connector Name
-iname      Template Internal Name
-tname      Template Name Displayed
-ttype      Template Type
             <templateType> 1 - inbound transformation
             <templateType> 2 - outbound transformation
             <templateType> 3 - XML outbound transformation
-d          Description
```

Example 2-1 Template Registration

The following command adds a new outbound template (`newTemplate.xml`) for a connector instance with the name Jira Connector Instance:

```
emctl register_template connector
-t <template.xml>
-repos_pwd <repos password>
-cname "Jira Connector Instance"
-iname "newTemplate"
-tname "newTemplateOutbound"
-ttype 2
-d "Demo template"
```

Viewing Template Code

To view template code, do the following:

1. Select the connector instance and go to the *Configure Management Connector* page.
2. Select the **Template** tab, as shown in the figure below.
3. Click a template name or click **View** to view the XSLT code for the template.

The ticket templates are in XSLT format. A basic knowledge of XSLT is required to understand the code.

Management Connectors
Management Connectors > Configure Management Connector: Jira Demo
Configure Management Connector: Jira Demo

General | **Template**

The following is the list of templates used by this connector.

Select	Name	Type	Description
<input checked="" type="radio"/>	Create Ticket Response	Inbound	This template handles responses from requests to create/update tickets in Jira
<input type="radio"/>	Get Ticket Response	Inbound	This template handles responses from retrieving tickets in Jira
<input type="radio"/>	Jira Bug Create and Update	Outbound	This template generates a request to create/update bugs in Jira
<input type="radio"/>	Jira Incident Create and Update	Outbound	This template generates a request to create/update tasks in Jira
<input type="radio"/>	Jira Problem Create and Update	Outbound	This template generates a request to create/update tasks in Jira
<input type="radio"/>	Jira Service Request Create and Update	Outbound	This template generates a request to create/update tasks in Jira
<input type="radio"/>	Jira Story Create and Update	Outbound	This template generates a request to create/update stories in Jira
<input type="radio"/>	Jira Subtask Create and Update	Outbound	This template generates a request to create/update subtasks in Jira
<input type="radio"/>	Jira Task Create and Update	Outbound	This template generates a request to create/update tasks in Jira
<input type="radio"/>	Get Ticket Request	XML Outbound	This template generates a request to retrieve a ticket in Jira

Removing a Template

To remove a template, do the following:

1. Select the connector instance and go to the *Configure Management Connector* page.
2. Select the **Template** tab.
3. Select the template and click **Delete**.
4. Click **Yes** to confirm the delete operation.

Editing Templates

To edit an existing ticket template, do the following:

1. Select the connector instance and go to the *Configure Management Connector* page.
2. Select the **Template** tab.
3. Select the template and click **Edit**.
4. Edit the Description and Template Contents fields as needed, then click **Save**.

The ticket template is automatically registered. Saving the registered template updates it in the repository.

Restoring Templates

After you edit or customize a template, you can restore it to its previous default contents if needed by performing the following steps.

1. Select the connector instance and go to the Configure Management Connector page.
2. Select the **Template** tab.
3. Select the template and click **Restore**.
4. Click **Yes** in the confirmation page that appears in order to restore.

Deleting the Connector

You can delete a connector from the Management Connector page by selecting the connector and clicking the **Delete** button. After a connector is deleted, all associated incident rules are no longer configured for ticketing operations.

Additionally, if the connector deleted is the only connector instance of that connector type, the corresponding connector type is also deleted.



Note:

The above rule is not applicable to the out-of-box connector type.

3

Creating Jira Tickets

You can create tickets automatically or manually. The following sections explain how to create both types.

- [Automatically Creating a Ticket](#)
- [Manually Creating a Ticket](#)



Note:

At least one ticketing connector must be configured and enabled in order to create a ticket using either method.

Automatically Creating a Ticket

To automatically create a ticket in Jira, you need to create a rule that specifies the event or incident condition on which the ticket should be created. In this rule, you also specify the ticket template to be used with the Jira connector.

To create a rule to automatically create a ticket in Jira:

1. From the Enterprise Manager console, click **Setup**, then **Incidents** and **Incident Rules**.
2. Click **Create Rule Set**. On the Create Rule Set page, enter a name and description.
3. Click the **Rules** tab.

As you proceed below, the wizard interface explains various rules and their usage. For detailed information, refer to the Using Incident Management chapter in the *Oracle Enterprise Manager Cloud Control Administrator's Guide*.

- For **events** —
This is the typical option to choose to create a ticket for an event, such as a *target down* availability event.
Click **Create** and select **Incoming events or updates to events** as a type of rule to create from the pop-up window. Follow the steps below to create a rule.
 - For **incidents** —
Click **Create** and select **Newly created incidents or updates to incidents** as a type of rule to create from the pop-up window. Follow the steps below to create a rule.
4. While configuring the above rule under the Add Actions page:
 - For **events** —
Select the **Create Incident** check box. This makes the Create Ticket option visible. Then, select the **Create Ticket** check box. It will display the combo box to select Connector Type, name, and template.
 - For **incidents** —

You can use this option to create tickets based on incident attributes, such as *priority 1* incidents or incidents based on fatal or critical severity.

Select the **Create Ticket** check box. It will display the combo box to select Connector Type, name, and template.

The following figures show the various Add Actions pages (events and incidents).

Add Actions Page for Events

Add Actions

Add Conditional Actions

Define actions to be taken when an event matches this rule.

Conditions for actions
You can define the actions to apply whenever the rule matches or apply them conditionally.

Always execute the actions
 Only execute the actions if specified conditions match

Create Incident or Update Incident
If there is no incident associated with the event, you could create one and optionally, set the incident owner and priority. If an incident exists, you could update the incident.

Create Incident (If not associated with one) Update Incident

Each event creates a new incident
 Compress events into an incident

Initial Setup

Create Ticket

Create Ticket - Select a ticketing device that you want to use to create a ticket for this incident.

Connector Type: Jira Service Management REST Ticketing Connector 13.5.0.0.0 ▼
Connector Name: Jira Demo ▼
Template Name: Jira Bug Create and Update
 Jira Incident Create and Update
 Jira Problem Create and Update
 Jira Service Request Create and Update
 Jira Story Create and Update
 Jira Subtask Create and Update
 Jira Task Create and Update

Send Notifications

Submit Corrective Action
Select a corrective action to corrective action is allowed.

Corrective action will use preferred credentials of SYSMAN (rule set owner) to execute scripts on respective targets.

Clear events
For most events, Enterprise Manager detects when the underlying issue is cleared and will generate a clear event. These types of events cannot be cleared using this option. However, if the underlying issue is cleared, this type of events must be manually cleared by administrators. This action can be used to automate this behavior.

Clear permanently

Forward to Event Connectors
Events can be forwarded to third party event management systems.
No event connector has been configured.

Add Actions Page for Incidents

Add Actions

Add Conditional Actions

Define actions to be taken when an incident matches this rule.

▲ **Conditions for actions**

You can define the actions to apply whenever the rule matches or apply them conditionally.

Always execute the actions

Only execute the actions if specified conditions match i

▶ **Send Notifications**

▲ **Update Incident**

Automatically update the incidents to help triage your incoming incidents.

Assign to

Set priority to Low ▼

Set status to None ▼

Escalate to ▼

▲ **Create Ticket**

Create Ticket - Select a ticketing device that you want to use to create a ticket for this incident.

Connector Type Jira Service Management REST Ticketing Connector 13.5.0.0.0 ▼

Connector Name Jira Demo ▼

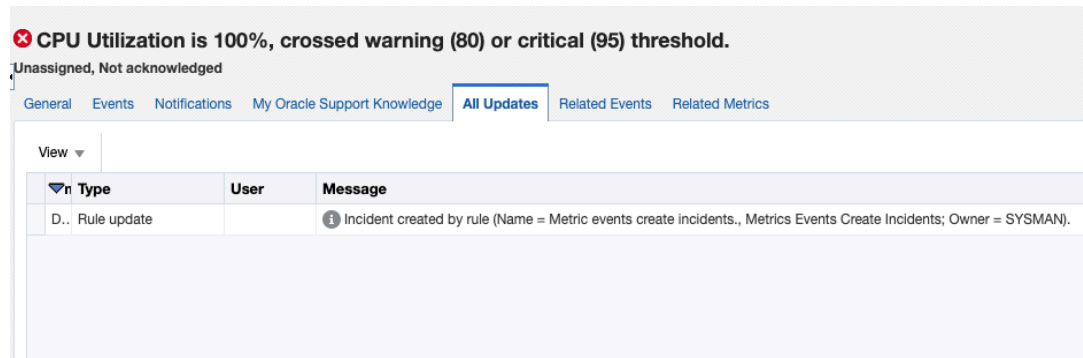
Template Name ✓ Jira Bug Create and Update

- Jira Incident Create and Update
- Jira Problem Create and Update
- Jira Service Request Create and Update
- Jira Story Create and Update
- Jira Subtask Create and Update
- Jira Task Create and Update

5. After selecting an appropriate connector and template, click **Save**.

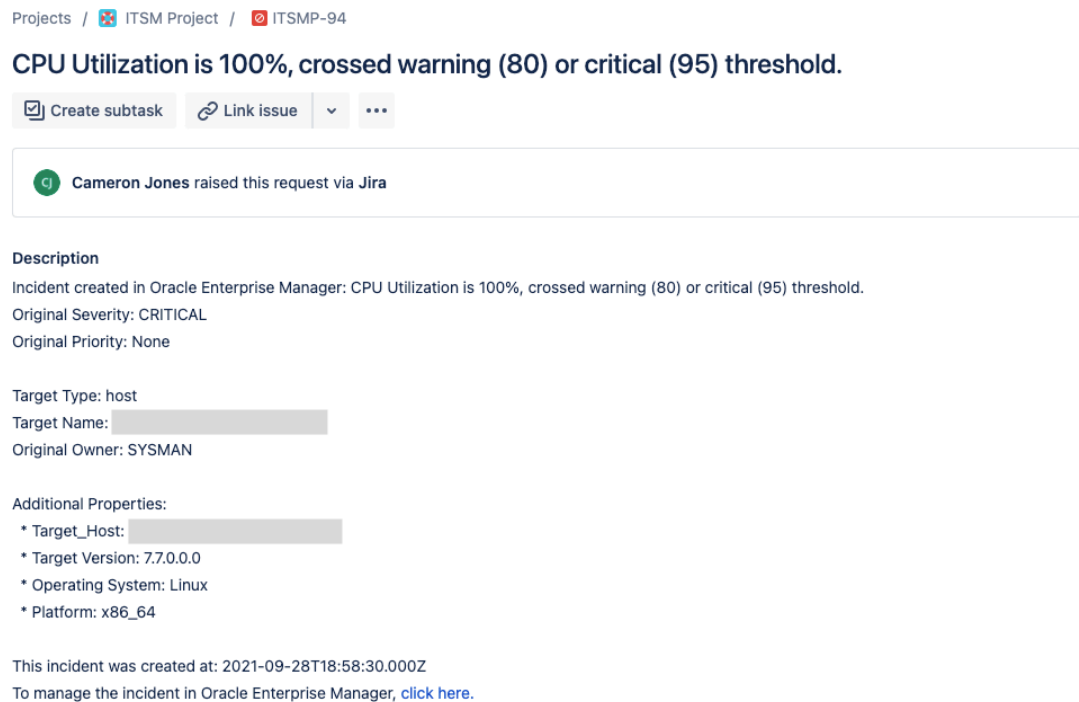
The following process occurs after you create the event or incident rule:

- A request is sent to the Jira Connector when a incident is created due to an event that matches your rule.
- The ticket is created or updated on the Jira system.
- In Enterprise Manager, the incident is annotated with a comment on the **All Updates** tab to indicate that a ticket was created or updated, as shown in the following figure:



Additionally, the incident is updated with the ticket ID, ticket status, and ticket URL under the Incident Tracking section.

A ticket is updated if there is an existing active ticket for an incident. The following figure shows the ticket in Jira.



Manually Creating a Ticket

Perform the following steps to manually create a ticket:

1. From the Enterprise menu of the Enterprise Manager console, select **Monitoring**, then **Incident Manager**.

The Incident Manager page appears, showing all open incidents.

2. Select any incident from the list in the table.
3. In the Tracking section, click **More**, then **Create Ticket**, as shown in the following figure.

Severity	Summary	Target	Priority	Status	Age	Time Since Last Update	Owner	Ackn	Escal	Type	Category
🔴	CPU Utilization is 100%, crossed warning (80) or critical (95) threshold.		None	New	Less than a min...	Less than a min...	-	No	No	Incident	Capacity
🟡	Coherence Cache is Down	/EMGC_GCDomain/GCDomain/def...	None	New	20 days ...	20 days 20 hours	-	No	No	Incident	Availability
🟡	Coherence Cache is Down	/EMGC_GCDomain/GCDomain/def...	None	New	20 days ...	20 days 20 hours	-	No	No	Incident	Availability

Rows Selected 1 Columns Hidden 23 Row count

🔴 CPU Utilization is 100%, crossed warning (80) or critical (95) threshold. [Open in new tab](#)

Unassigned, Not acknowledged

General Events Notifications My Oracle Support Knowledge All Updates Related Events Related Metrics

Incident Details

ID 1382

Metric CPU Utilization (%)

Metric Group Load

Target (host)

Incident Created Oct 11, 2021 2:15:30 PM GMT

Last Updated Oct 11, 2021 2:15:30 PM GMT

Summary CPU Utilization is 100%, crossed warning (80) or critical (95) threshold.

Internal Event Name Load:cpuUtil

Event Type Metric Alert

Category Capacity

Show internal values for attributes ...

Tracking

Escalated No

Priority None

Status New

Owner -

Acknowledged No

Last Incident created by rule (Name = Incident management rule set for all targets. Create incident...)

Comment [System generated rule]: on Oct 11, 2021 2:15:30 PM GMT

This incident will be automatically cleared when the underlying issue is resolved.

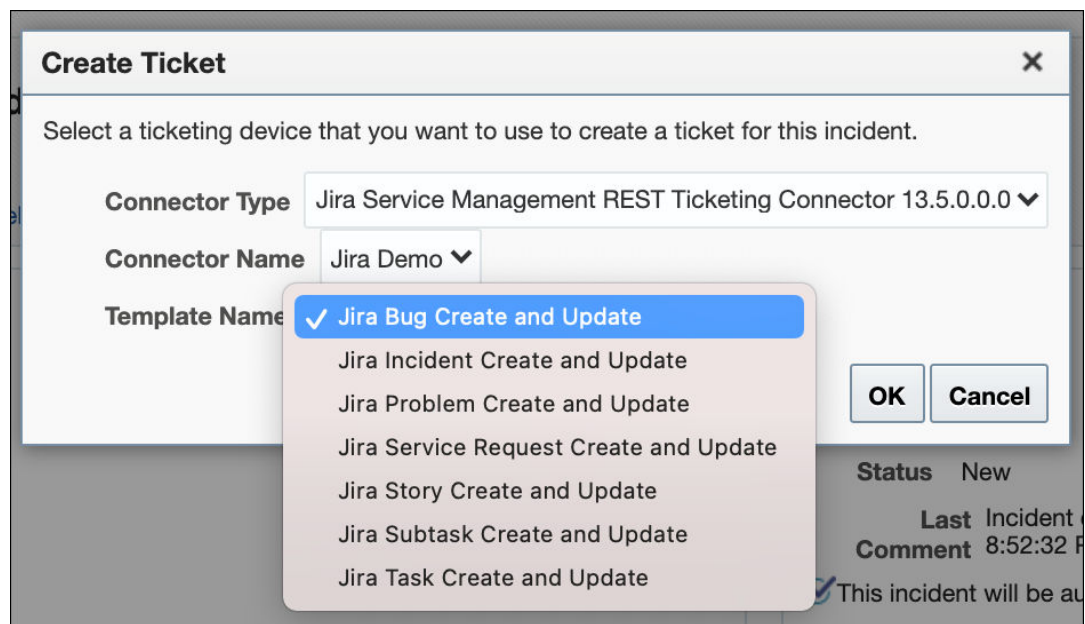
Guided Resolution

Diagnostics View system load and top processes Problem Analysis View Metric Help

Actions Reevaluate Alert Edit Thresholds

Corrective Actions No corrective action defined. Add corrective action

- In the Create Ticket pop-up window, select a ticket template from the Template Name drop-down menu as shown in the following figure.



Click **OK**.

If you do not see the desired template, you can add one using the `emctl` command. See [Adding Ticket Templates](#).

If creating ticket is successful, the ticket ID appears in the tracking section for the incident. If the Web console settings are configured and enabled, the ticket ID appears as a link to the ticket page in Jira.

4

Using Default Templates

This chapter provides details on the default ticket templates shipped along with the Jira Connector. The ticket templates specify the mappings between an Enterprise Manager incident and underlying event attributes and Jira ticket attributes.

- [Template Process](#)
- [Reading Ticket Templates](#)
- [Customizing Ticket Templates](#)
- [Defining New Templates](#)

Template Process

All out-of-box templates cause the following actions to occur when you create a ticket for an incident:

1. Write incident information to `Description` (Jira ticket description).
2. Set the Jira ticket title based on the incident message.

The out-of-box templates are as follows:

- `jiraStoryCreateAndUpdate.xml`
This template will create Oracle Enterprise Incidents as a Story ticket in Jira.
- `jiraBugCreateAndUpdate.xml`
This template will create Oracle Enterprise Incidents as a Bug ticket in Jira.
- `jiraTaskCreateAndUpdate.xml`
This template will create Oracle Enterprise Incidents as a Task ticket in Jira.
- `jiraSubtaskCreateAndUpdate.xml`
This template will create Oracle Enterprise Incidents as a Subtask ticket in Jira.
- `jiraIncidentCreateAndUpdate.xml`
This template will create Oracle Enterprise Incidents as an Incident ticket in Jira.
- `jiraProblemCreateAndUpdate.xml`
This template will create Oracle Enterprise Incidents as a Problem ticket in Jira.
- `jiraServiceRequestCreateAndUpdate.xml`
This template will create Oracle Enterprise Incidents as a Service Request ticket in Jira.

Reading Ticket Templates

The following table illustrates the creation of a ticket using `jiraStoryCreateAndUpdate.xml`. This illustration will help you to read a ticket template. In the table, * denotes a literal string and ** indicates if the attribute applies.. Additionally, the table also shows Jira ticket attributes and corresponding Enterprise Manager incident values for ticket creation mappings.

Ticket creation mappings are the same for `jiraStoryCreateAndUpdate.xsl`, `jiraBugCreateAndUpdate.xsl`, `jiraTaskCreateAndUpdate.xsl`, and `jiraSubtaskCreateAndUpdate.xsl`.

Table 4-1 Ticket Creation Mappings (for all templates)

Jira Ticket Attributes	Enterprise Manager Incident Attributes	Value
Issue type	.	10001 (this is the <i>story</i> type and changes between templates)
project/key	\$PROJECT_ID\$	Set to the value of Project Key during configuration.
labels	.	"oracle_enterprise_manager" *
description	Summary	Summary – The details about the incident that has fired
	Severity Code	Severity Code – The severity of the incident in Oracle Enterprise Manager
	Priority	Priority – The priority of the incident in Oracle Enterprise Manager
	Notification Rule Name	Notification Rule Name – Name of the rule that created the ticket (Auto ticketing only)
	Notification Rule Owner	Notification Rule Owner – Name of the user that owns the rule that created the ticket (Auto ticketing only)
	Target Type	Target Type – The type of target that the incident fired on
	Target Name	Target Name – The name of the target that the incident fired on
	Obj Owner	Obj Owner – Name of the user that owns the object the incident is fired on
	Target Properties	Target Properties – Additional names and values about the incident. Includes things like host, version, OS and platform
	Last Updated Date	Last Updated Date – Date the incident was created in Oracle Enterprise Manager
Incident URL	Incident URL – A URL link that directs the user to the Oracle Enterprise Manager Incident	
Title	Summary	Summary – The details about the incident that has fired New *

Use the mapping table above as a reference to read the following XSLT file (`jiraStoryCreateAndUpdate.xsl`).

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:emcf="http://
xmlns.oracle.com/sysman/connector" version="1.0">
  <xsl:template match="emcf:EMIncident">
    <!-- Please remember to backup all files before editing -->
    <!--
      This area is where variables are set to be used in the rest of the template.
      By default, there are two variables set:
      * Action: Used to figure out if the connector is 'creating' or 'updating' the
      Jira Ticket.
      * Severity: Used to set the Severity of the Jira ticket (if used).
      * Priority: Used to set the Priority of the Jira ticket.

      If you want to add or change variables, do so below.
    -->
    <xsl:variable name="action">
      <xsl:choose>
        <xsl:when test="normalize-space(emcf:TicketID) = ''">CREATE</xsl:when>
        <xsl:otherwise>UPDATE</xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
  </xsl:template>
</xsl:stylesheet>
```



```

    </xsl:variable>
    <xsl:variable name="severity">
      <xsl:choose>
        <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'FATAL'">Sev-0</
xsl:when>
        <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'CRITICAL'">Sev-0</
xsl:when>
        <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'WARNING'">Sev-1</
xsl:when>
        <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'ADVISORY'">Sev-2</
xsl:when>
        <xsl:otherwise>Sev-3</xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <xsl:variable name="priority">
      <xsl:choose>
        <xsl:when test="emcf:SystemAttributes/emcf:Priority = 'Urgent'">Highest</
xsl:when>
        <xsl:when test="emcf:SystemAttributes/emcf:Priority = 'Very High'">Highest</
xsl:when>
        <xsl:when test="emcf:SystemAttributes/emcf:Priority = 'High'">High</xsl:when>
        <xsl:when test="emcf:SystemAttributes/emcf:Priority = 'Medium'">Medium</xsl:when>
        <xsl:when test="emcf:SystemAttributes/emcf:Priority = 'Low'">Low</xsl:when>
        <xsl:otherwise>Medium</xsl:otherwise>
      </xsl:choose>
    </xsl:variable>
    <Message>
      <!--
        The following areas (Method, Variables, and Headers) should not be edited.
        They are used to make communication work between Oracle Enterprise Manager and
Jira.
      -->
    <Method>
      <xsl:choose>
        <xsl:when test="$action = 'CREATE'">POST</xsl:when>
        <xsl:when test="$action = 'UPDATE'">POST</xsl:when>
        <xsl:otherwise>ERROR</xsl:otherwise>
      </xsl:choose>
    </Method>
    <Variables>
      <Variable>
        <Name>TKT_KEY</Name>
        <Value>
          <xsl:value-of select="normalize-space(emcf:TicketID)" />
        </Value>
      </Variable>
    </Variables>
    <Headers>
      <Header>
        <Name>Authorization</Name>
        <Value>Basic $TOKEN$</Value>
      </Header>
      <Header>
        <Name>Accept</Name>
        <Value>*/*</Value>
      </Header>
    </Headers>
    <!--
      The body is used to send data to Jira. By default, the connector sends data
either 'On Create' or 'On Update'.
    -->

```

Fields that are sent when the ticket is created are found under the following

```
header: <xsl:if test="$action = 'CREATE'">
  * Summary: This is the title of Jira ticket
  * Issue Type: This is the type of the Jira ticket. By default, it is set to
'Story'
  * Project: This is the project the Jira tickets will be created under. This is
taken from the configuration settings.
  * Labels: This is an array of any labels you wanted added to the ticket.
  * Description: This is the description of the Jira ticket.
  * Priority: This is the priority of the Jira ticket based on the priority of
the Oracle Enterprise Manager ticket.
```

Fields that are sent when the ticket is updated are found under the following

```
header: <xsl:if test="$action = 'UPDATE'">
  Note: These items will be sent as a 'comment' in Jira
  * Body: This is the description of current values when an incident is updated
in Oracle Enterprise Manager
```

There are also a number of fields that are not enabled in Jira by default. They are included but commented out.

To see if you have these fields enabled, go to your Jira project > 'Project Settings' > 'Issue Types' > 'Story'

If your Jira instance uses these fields, feel free to uncomment them and your tickets will be created with them as well.

Note: If you uncomment a field that your Jira project is not using, it will fail to create the ticket.

Additionally, there are a number of fields that Jira has provided or that you have created called 'custom fields'.

To find your custom fields navigate to 'Administration (gear icon)' > 'Issues' > 'Custom Fields'.

To find the ID of your custom field, click on the '...' and select 'Edit Details' or 'View field information'.

Once you are on this page, the ID will be in the URL.

Examples:

```
* customFieldId=10000
* id=10004
```

Once you have your ID, you can add a custom field with the name customfield_<ID>.

Here are a few examples with some different types of custom fields:

Let's say we wanted to add the 'requested participants' field in Jira and found it's ID was '30032'

Requested participants is an array of users. To add it to the fields, we would add the following:

```
<array name="customfield_30032">
  [
    <object>
      <string name='id'>"USER ID 1"</string>,
    </object>
    <object>
      <string name='id'>"USER ID 2"</string>
    </object>
  ]
</array>
```

Let's say we wanted to add the 'severity' field in Jira and found it's ID was '30043'

Severity is an object with a value. To add it to the fields, we would add the following:

```
<object name="customfield_30043">
  <string name="value">"SEV-0"</string>
</object>
```

If you don't want all your issues to be SEV-0 in Jira, use the \$severity variable above to dynamically set severity:

```
<object name="customfield_30043">
  <string name="value">
    <xsl:value-of select="$severity"/>
  </string>
</object>
```

Finally, let's say you wanted to add a component to Jira based on Target Type in Oracle Enterprise Manager. This would require a few steps.

First, we need to set a component variable. This variable will be located above with the other variables (action, severity and priority).

Here is an example of what that variable would look like:

```
<xsl:variable name="component">
  <xsl:choose>
    <xsl:when test="emcf:SystemAttributes/emcf:SourceInfo/emcf:TargetInfo/
emcf:TargetType = 'host'">Host</xsl:when>
    <xsl:when test="emcf:SystemAttributes/emcf:SourceInfo/emcf:TargetInfo/
emcf:TargetType = 'agent'">Agent</xsl:when>
    <xsl:otherwise>Default</xsl:otherwise>
  </xsl:choose>
</xsl:variable>
```

In this example you can see that we look if the target type is a 'host' or 'agent' and returns the component name based on that ('Host' or 'Agent').

If it does not find either target type, it returns a default value of 'Default'.

If you would like to add more target types, simply copy one of the 'when' statements and change the target type and return value.

Now, that the variable is set, we must verify that our return values are valid components in Jira.

To do that, go to your Jira Project > 'Project Settings' > 'Components'

If the component does not exist, click 'Create component' and create it. In this example, we created 'Host', 'Agent', and 'Default'.

Finally, in the body below, we need to add component. It needs to be added under the line: <object name="fields">

Here is a good example of what that should look like:

```
<array name="components">
  [
    <object>
      <string name="name"><xsl:value-of select="$component"/></string>
    </object>
  ]
</array>
-->
<Body>
  <object>
    <xsl:if test="$action = 'CREATE'">
      <object name="fields">
        <object name="issuetype">
          <string name="name">Story</string>
        </object>
        <!-- This is the title of the Jira ticket. -->
        <string name="summary">
          <xsl:value-of select="emcf:SystemAttributes/emcf:Summary" />
        </string>
        <!-- This is the Project that the ticket is created under in Jira. This is
taken from the configuration settings. -->
        <object name="project">
          <string name="key">${PROJECT_ID$}</string>
```

```

        </object>
        <!-- This is an array of labels applied to the ticket in Jira. By default,
we added the label 'oracle_enterprise_manager' -->
        <array name="labels">
            [
                <string>oracle_enterprise_manager</string>
            ]
        </array>
        <!-- This is the main description of the ticket in Jira. -->
        <object name="description">
            <string name="type">doc</string>
            <number name="version">1</number>
            <array name="content">
                [
                    <object>
                        <string name="type">paragraph</string>
                        <array name="content">
                            [
                                <object>
                                    <string name="text">Incident created in Oracle Enterprise
Manager: <xsl:value-of select="emcf:SystemAttributes/emcf:Summary" />
Original Severity: <xsl:value-of select="emcf:SystemAttributes/emcf:SeverityCode" />
Original Priority: <xsl:value-of select="emcf:SystemAttributes/emcf:Priority" />
<xsl:choose>
                    <xsl:when test="normalize-space(emcf:NotificationRuleName) != ''">
Incident triggered by the following rule: <xsl:value-of
select="emcf:NotificationRuleName" />
                    </xsl:when>
                </xsl:choose>
                <xsl:choose>
                    <xsl:when test="normalize-space(emcf:NotificationRuleOwner) != ''">
Rule is owned by: <xsl:value-of select="emcf:NotificationRuleOwner" />
                    </xsl:when>
                </xsl:choose>

Target Type: <xsl:value-of select="emcf:SystemAttributes/emcf:SourceInfo/emcf:TargetInfo/
emcf:TargetType" />
Target Name: <xsl:value-of select="emcf:SystemAttributes/emcf:SourceInfo/emcf:TargetInfo/
emcf:TargetName" />
Original Owner: <xsl:value-of select="emcf:SystemAttributes/emcf:SourceInfo/
emcf:SourceObjInfo/emcf:ObjOwner" />

Additional Properties: <xsl:for-each select="emcf:SystemAttributes/emcf:SourceInfo/
emcf:TargetInfo/emcf:TargetProperty">
    * <xsl:value-of select="./emcf:Name" />: <xsl:value-of select="./emcf:Value" />
</xsl:for-each>

This incident was created at: <xsl:value-of select="emcf:SystemAttributes/
emcf:LastUpdatedDate" />
To manage the incident in Oracle Enterprise Manager, </string>
            <string name="type">text</string>
        </object>
        <object>
            <string name="text">click here.</string>
            <string name="type">text</string>
            <array name="marks">
                [
                    <object>
                        <string name="type">link</string>
                        <object name="attrs">
                            <string name="href">
                                <xsl:value-of select="emcf:SystemAttributes/

```

```

emcf:IncidentURL" />
        </string>
      </object>
    </object>
  ]
</array>
</object>
]
</array>
</object>
]
</array>
</object>
<!-- Sets Priority in your Jira Issue. -->
<object name="priority">
  <string name="name">
    <xsl:value-of select="$priority" />
  </string>
</object>
<!-- This will put the Jira Issue into a component (or subsection).
Replace 'COMPONENT NAME-X' with the component names you wish the Issue to be a part of.
-->
<!--
  <array name="components">
    [
      <object>
        <string name="name">COMPONENT NAME-1</string>
      </object>
      <object>
        <string name="name">COMPONENT NAME-2</string>
      </object>
    ]
  </array>
-->
<!-- This will set the reporter of the Jira Issue. Replace 'REPORTER ID'
with the User's ID. -->
<!--
  <object name="reporter">
    <string name="id">REPORTER ID</string>
  </object>
-->
<!-- This will set the version of the release this Issue should be
completed in. Replace 'FIXVERSION NAME-X' and implement the versions this should be
worked on. -->
<!--
  <array name="fixVersions">
    [
      <object>
        <string name="name">FIXVERSION NAME-1</string>
      </object>
      <object>
        <string name="name">FIXVERSION NAME-2</string>
      </object>
    ]
  </array>
-->
<!-- This will set the remaining time and the original estimated time for
the Jira Issue. Replace '100' with the number of minutes for each estimate. -->
<!--
  <object name="timetracking">
    <string name="remainingEstimate">100</string>
    <string name="originalEstimate">100</string>
  </object>
-->

```

```

    </object>
  -->
  <!-- This will set environment details in the Jira Issue. By default, it
will link to the Target URL inside Oracle Enterprise Manager. To add more details,
implement them under 'text'. -->
  <!--
  <object name="environment">
    <string name="type">doc</string>
    <number name="version">1</number>
    <array name="content">
      [
        <object>
          <string name="type">paragraph</string>
          <array name="content">
            [
              <object>
                <string name="text">To visit the target in Oracle Enterprise
Manager, </string>
                <string name="type">text</string>
              </object>
              <object>
                <string name="text">click here.</string>
                <string name="type">text</string>
                <array name="marks">
                  [
                    <object>
                      <string name="type">link</string>
                      <object name="attrs">
                        <string name="href"><xsl:value-of
select="emcf:SystemAttributes/emcf:SourceInfo/emcf:TargetInfo/emcf:TargetURL"/></string>
                      </object>
                    </object>
                  ]
                </array>
              </object>
            ]
          </array>
        </object>
      ]
    </array>
  </object>
  -->
  <!-- This will set the affected version the Jira Issue was found on.
Replace 'VERSION NAME-X' with the version name. -->
  <!--
  <array name="versions">
    [
      <object>
        <string name="name">VERSION NAME-1</string>
      </object>
      <object>
        <string name="name">VERSION NAME-2</string>
      </object>
    ]
  </array>
  -->
  <!-- This will set the date the Jira issue is due by. Replace YYYY-MM-DD
with the Year, Month and Day. -->
  <!--
  <string name="duedate">YYYY-MM-DD</string>
  -->
  <!-- This will assign the Jira Issue to a user. Replace 'ASSIGNEE ID' with

```

```

the User's ID. -->
    <!--
      <object name="assignee">
        <string name="id">ASSIGNEE ID</string>
      </object>
    -->
  </object>
</xsl:if>
<xsl:if test="$action = 'UPDATE'">
  <object name="body">
    <string name="type">doc</string>
    <number name="version">1</number>
    <array name="content">
      [
        <object>
          <string name="type">paragraph</string>
          <array name="content">
            [
              <object>
                <string name="text">The incident was updated in Oracle Enterprise
Manager.
<xsl:value-of select="emcf:SystemAttributes/emcf:Summary" />

Severity is currently: <xsl:value-of select="emcf:SystemAttributes/emcf:SeverityCode" />
Priority is currently: <xsl:value-of select="emcf:SystemAttributes/emcf:Priority" />
Owner is currently: <xsl:value-of select="emcf:SystemAttributes/emcf:SourceInfo/
emcf:SourceObjInfo/emcf:ObjOwner" />
This incident was updated at: <xsl:value-of select="emcf:SystemAttributes/
emcf:LastUpdatedDate" />
                </string>
                <string name="type">text</string>
              </object>
            ]
          </array>
        </object>
      ]
    </array>
  </object>
</xsl:if>
</Body>
</Message>
</xsl:template>
</xsl:stylesheet>

```

Customizing Ticket Templates

If the out-of-box ticket templates do not satisfy your requirements, you can modify them. To do this, Oracle recommends that you use one of the existing templates as the base template. Copy this ticket template to a new file, modify, and register the new ticket template.

You can edit a template directly on Enterprise Manager instead of using the EM CLI command. See [Editing Templates](#) and [Restoring Templates](#) for more information.

In most cases, when you modify the ticket template, you might only be changing the mappings. The following examples illustrate this concept:

Example 4-1 Add Due Date to Jira Ticket

To create a template to have a due date in Jira, find the following code in the template and remove the comment brackets (<!-- and -->) and set the date:

```
<!--
    <string name="duedate">2023-02-17</string>
-->
```

Example 4-2 Add a custom field into the template

Jira allows you to create your own custom fields. To add a custom field of your own, find the ID in Jira and add it into the template. To find your custom fields, from the **Administration** (gear icon) menu, select **Issues** and then **Custom Fields**. To find the ID of your custom field, click on the ellipses (...) and select **Edit Details** or **View field information**. Once you are on this page, the ID will be in the URL. Once you have your ID, you can add a custom field with the name `customfield_<ID>`. Here is an example of a custom string field:

```
<string name="customfield_20000">"My custom field value"</string>
```

The templates are highly customizable. Oracle recommends that only users with advanced knowledge of XSLT make complex changes.

You can use incident or event rules as a filter to associate proper ticket templates with incidents. You can have as many tickets templates as desired. One incident or event rule can have only one ticket template.

Defining New Templates

The out-of-box templates are based on the default forms. If the new ticket templates you define are based on these forms, [Customizing Ticket Templates](#) applies.

However, if you use a different form, you need to define a new ticket template.

Enterprise Manager Attributes

The following table provides the Enterprise Manager fields that you can map when using Jira:

Table 4-2 Enterprise Manager Attributes

Data Fields	Description
EMUser	<ul style="list-style-type: none"> For auto-ticketing, this is the incident rule owner. For manual ticketing, this is the console user that triggered the ticket creation.
TicketID	Identifies the ticket associated with the current incident (available after ticket creation).
TargetType	Type of target that the incident is associated with, such as <code>host</code> .
TargetName	Name of the target that the incident is associated with. For example, <code>Databasel</code> or <code>yourhost.example.com</code> .

Table 4-2 (Cont.) Enterprise Manager Attributes

Data Fields	Description
TargetProperties	TargetProperties store environmental or usage context information specific to the target. For example, for Host Target, the name /value pair of TargetProperties are: <ul style="list-style-type: none"> • Comment — Host running the management service and repository. • Contact — John Doe • Deployment Type — Production • Line of Business — Development • Location — Redwood Shores These are out-of-box user-defined target properties. If additional target properties are added, they are displayed with ticket information.
Summary	Description of the incident. For example: CPU Utilization is 100%, crossed warning (80) or critical (95) threshold.
Priority	Priority of the incident from low to urgent.
Severity	Severity of the incident: critical, warning, clear, or down.
CollectionTime	Timestamp of an incident occurrence.
IncidentURL	URL to the incident details page of the incident.
NotificationRuleName	Name of the incident rule that generated the notification during auto-ticketing.
TargetTimezone	Timezone of the target associated with the incident.

Format for Creating Ticket Templates

To create ticket templates for custom Jira forms, adhere to the following format:

Example 4-3 Template Format for Custom Jira Forms

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:emcf="http://
xmlns.oracle.com/sysman/connector" version="1.0">
  <xsl:template match="emcf:EMIncident">
    * [Insert any variables needed here]
    <Message>
      <Method>
        <xsl:choose>
          <xsl:when test="$action = 'CREATE'">POST</xsl:when>
          <xsl:when test="$action = 'UPDATE'">POST</xsl:when>
          <xsl:otherwise>ERROR</xsl:otherwise>
        </xsl:choose>
      </Method>
      <Variables>
        <Variable>
          <Name>TKT_KEY</Name>
          <Value>
            <xsl:value-of select="normalize-space(emcf:TicketID)" />
          </Value>
        </Variable>
      </Variables>
      <Headers>
        <Header>
          <Name>Authorization</Name>
          <Value>Basic $TOKEN$</Value>
        </Header>
      </Headers>
    </Message>
  </template>
</xsl:stylesheet>
```

```
        </Header>
        <Header>
            <Name>Accept</Name>
            <Value>*</Value>
        </Header>
    </Headers>
    <Body>
* [Insert body changes (details you are sending to Jira) here]
    </Body>
</Message>
</xsl:template>
</xsl:stylesheet>
```

Index

C

- creating
 - trouble tickets manually (Jira), [3-4](#)
- customizing
 - Jira ticket templates, [4-9](#)

D

- default templates, using for Jira Connector, [4-1](#)

F

- format for creating
 - Jira ticket templates, [4-10](#)

J

- Jira Connector
 - customizing ticket templates, [4-9](#)

Jira Connector (*continued*)

- default templates, using, [4-1](#)
- format for creating ticket templates, [4-10](#)
- replacing a template, [2-7](#)
- ticket templates, format for creating, [4-10](#)

R

- replacing a template
 - Jira Connector, [2-7](#)

T

- templates
 - customizing Jira ticket templates, [4-9](#)
 - replacing, [2-7](#)
- ticket templates
 - customizing Jira ticket templates, [4-9](#)
- trouble tickets
 - creating manually for Jira Connector, [3-4](#)