

Oracle® Enterprise Manager PagerDuty Connector Installation and Configuration Guide



24ai Release 1 (24.1)

F97236-01

December 2024

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

ORACLE®

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

F97236-01

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

Contents

Preface

Audience	v
Documentation Accessibility	v
Related Resources	v
Conventions	vi

1 Introduction to the Connector

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

2 Installing and Configuring the 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	2-9
Adding Ticket Templates	2-9
Viewing Template Code	2-10
Removing a Template	2-10
Editing Templates	2-11
Restoring Templates	2-11
Deleting the Connector	2-11

3 Creating PagerDuty 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	4-5
Defining New Templates	4-6

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://otn.oracle.com/membership/>

If you already have a user name and password for OTN, then you can go directly to the documentation section of the OTN Web site at:

<http://otn.oracle.com/documentation/>

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 Connector

Versions Supported

The PagerDuty Connector communicates through version 2 of the REST API component of PagerDuty 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 PagerDuty Incident 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 PagerDuty to close the ticket.

 **See Also:**

[Automatically Creating a Ticket](#)

Manual Ticketing

From the Enterprise Manager console, you can manually open a PagerDuty incident based on an open incident in Enterprise Manager. The PagerDuty 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 PagerDuty incident format before the requests are sent to PagerDuty. A ticket template specifies how an Enterprise Manager incident and its associated event attributes can be mapped to the ticket attributes of PagerDuty.

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 PagerDuty. For manual ticketing, you have to select a connector instance and ticket template before submitting a request for ticket creation.

The PagerDuty 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 recurrence 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 PagerDuty. If the grace period is one hour and the event is cleared at 10:00 AM, this clears the incident and ticket. If the same event reoccurs before 11:00 AM and enforces creation of another incident, the grace period functionality stops creation of a new ticket for this and reopens the same ticket instead. Note: This only works if the ticket is not resolved in PagerDuty.

 **Note:**

In PagerDuty, after a ticket is set to a Closed status, it cannot be reopened. Consequently, an incident that re-triggers within the grace period cannot reopen the ticket but only annotate it.

If you want to reopen a ticket for incident occurrences that fall within the grace period, set the ticket status to Resolved instead of Closed when the incident clears. This enables the PagerDuty Connector to reopen the ticket if the same incident reoccurs within the grace period.

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 Connector

This chapter provides the following information for setting up and configuring the PagerDuty 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 PagerDuty Connector communicates through PagerDuty's REST API. Before using the PagerDuty Connector, ensure that you meet the following prerequisites:

- PagerDuty 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 PagerDuty 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 Pre-requests 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.
Always-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 OAM Migration Utility	0	0	0	0 Utility to migrate assets managed by OAM 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 PagerDuty 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 PagerDuty connector appears in the list with a status of *Available*.
3. Click on the appropriate PagerDuty 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 PagerDuty 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 PagerDuty 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, Service ID and so forth of the PagerDuty instance. Multiple connector instances for the same connector type are supported, which provides the flexibility to configure multiple PagerDuty 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. .

The screenshot shows the Oracle Enterprise Manager console interface. At the top, it says "ORACLE Enterprise Manager Cloud Control 13c". Below that, there are navigation tabs for "Enterprise", "Targets", "Favorites", and "History". The main heading is "Management Connectors". A sub-heading explains that a Management Connector integrates different enterprise frameworks. Below this is a "Create Connector" section with a dropdown menu for "Select connector type" and a "Go" button. A table lists the available connectors:

Select	Status	Name	Type	Version	Category	Description
<input type="radio"/>		PagerDuty Demo	PagerDuty REST Connector	13.5.0.0.0	Ticketing Connector	PagerDuty Ticketing Connector

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 PagerDuty 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 PagerDuty Connector should have a check mark in the Status column, as shown in the following figure.

The screenshot shows the Oracle Enterprise Manager console interface. At the top, it says "ORACLE Enterprise Manager Cloud Control 13c". Below that, there are navigation tabs for "Enterprise". The main heading is "Management Connectors". A yellow banner with a checkmark icon says "Confirmation: Connection test succeeded. The configuration was saved." Below this is a sub-heading explaining that a Management Connector integrates different enterprise frameworks. Below this is a "Create Connector" section with a dropdown menu for "Select connector type" and a "Go" button. A table lists the available connectors:

Select	Status	Name	Type	Version	Category	Description
<input type="radio"/>	✓	PagerDuty Demo	PagerDuty REST Connector	13.5.0.0.0	Ticketing Connector	

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. See [Adding Ticket Templates](#) for more information about this command and template registration.

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 Oracle Enterprise Manager Cloud Control 13c interface for configuring a PagerDuty connector. The 'General' tab is selected, and the 'Connection Settings' section is expanded. It includes fields for 'Web Service End Points' (createTicket, getTicket, updateTicket), 'Token', 'User Email', 'Service ID', and 'Ticket ID'. There are also 'Proxy Settings' for Host, Port, Username, and Password. The 'Web Console Settings' section has a checkbox for 'Enable web console features' and a 'Pagerduty Subdomain' field. The 'Grace Period' section has a checkbox for 'Enable grace period' and a 'Grace Period (Hours)' field. The 'Retry' section has a checkbox for 'Enable Retry' and an 'Expiration (Hours)' field.

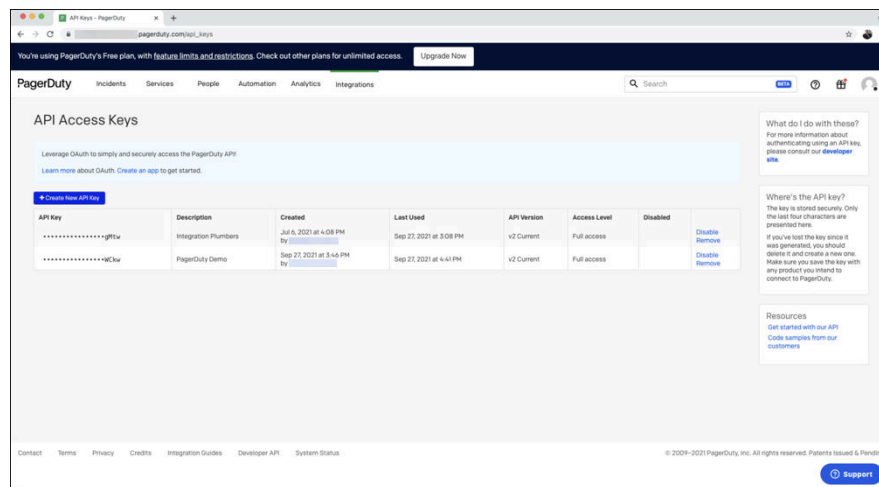
Connection Settings

The PagerDuty Connector communicates with the Service Desk through their Web services.

- **WebService End Points** —End points to createTicket,updateTicket,and

getTicketwebservice exposed by PagerDuty. These should not need to be edited.

- **Token ID**—API Token with permissions to retrieve, create, and update tickets from PagerDuty.
 1. Log in to PagerDuty.
 2. Hover over the **Integrations** tab.
 3. Under the Develop Tools section, select API Access Keys.
 4. On the landing page, click **Create new API Key** and follow the instructions.



Create API Key ✕

Description:

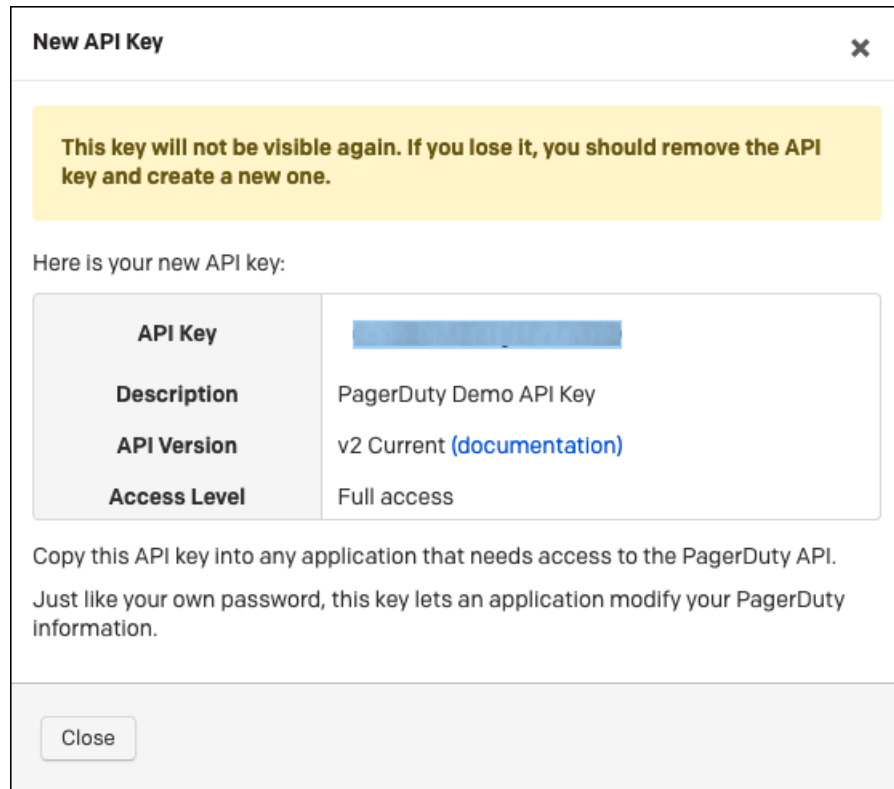
[API Documentation](#)

Read-only API Key ?

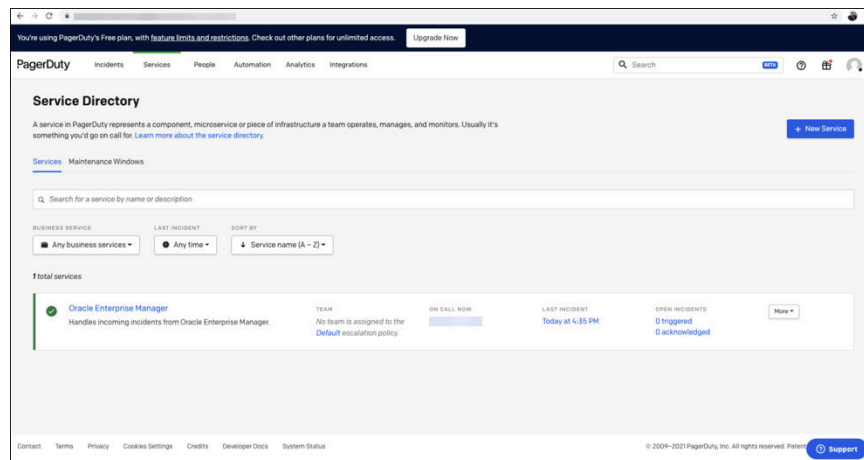
Create Key
Close

Note:

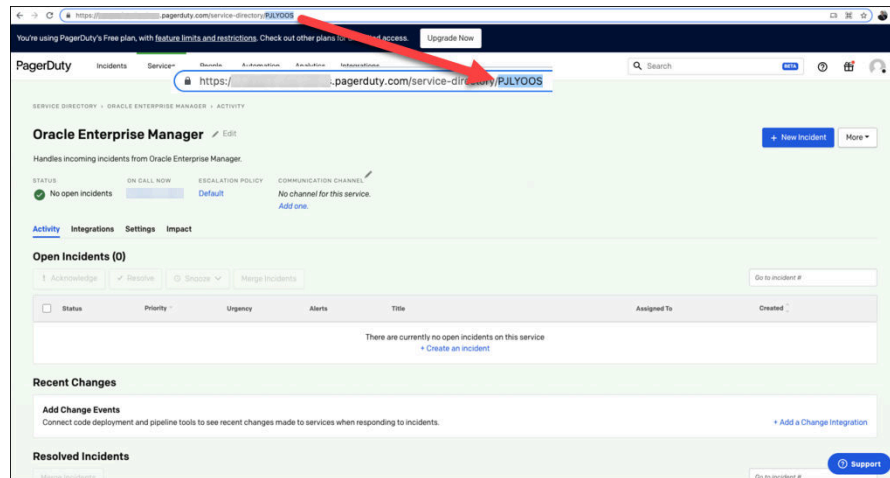
Leave the *Read-only API Key* option unchecked. You will be using the *Create API* to create tickets in PagerDuty.



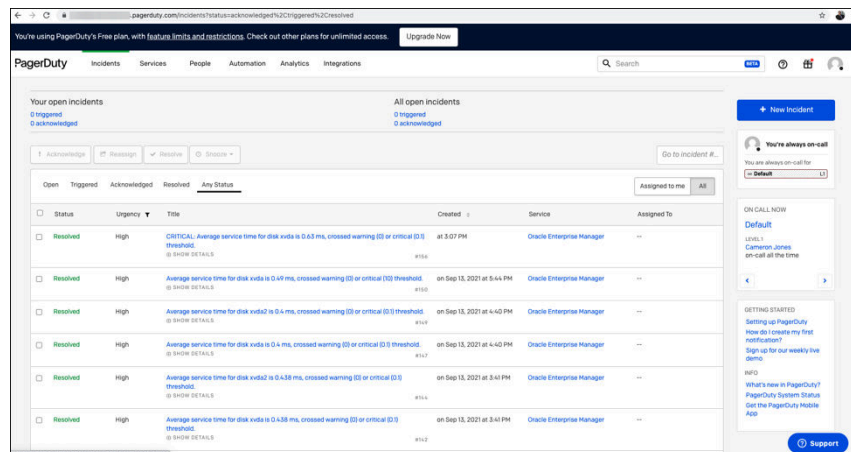
- **User Email**—Email of the PagerDuty user attached to the API Key.
- **Service ID**—The ID of the PagerDuty Service the incidents will be created in.
 1. Log in to PagerDuty.
 2. Hover over the **Services** tab.
 3. Select **Service Directory**.
 4. If you have not already done so, create a service for Oracle Enterprise Manager.
 5. Once your service is created, click on the service name.



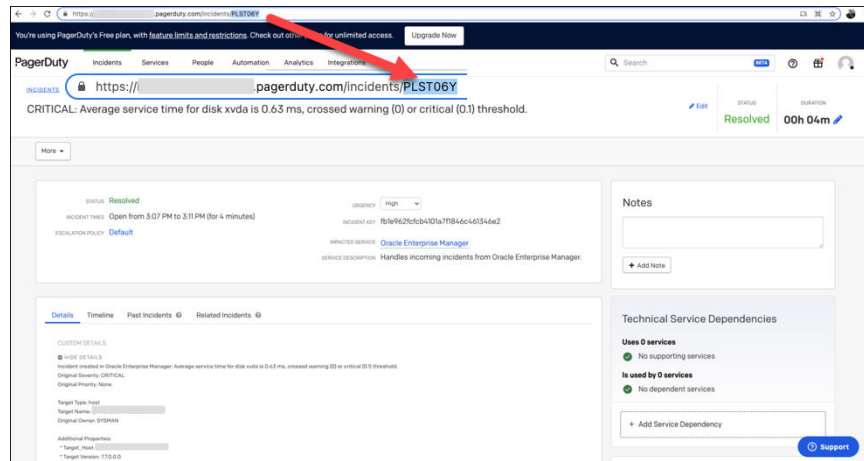
6. To get the Service ID, navigate to the landing page and copy the seven letter string at the end of the URL.



- **Ticket ID**—An existing Ticket ID to test the connection to the PagerDuty system. The ticket does not need to be open.
 1. Log in to PagerDuty.
 2. Click on the Incidents tab.
On the landing page, you'll see a list of your incidents.



3. From the list of incidents, click on the incident for which you want the Ticket ID. The individual ticket information displays.
4. To determine the Ticket ID of the selected incident, navigate to the URL and copy the seven letter string at the end of the URL.

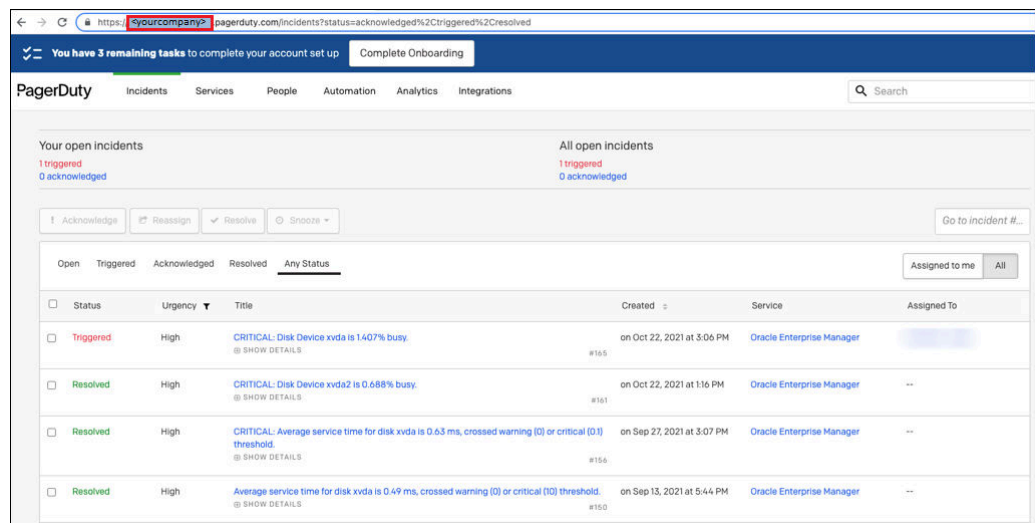


Web Console Settings

Web Console settings are required if you want the connector to provide links to PagerDuty 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 PagerDuty incident page within context from Enterprise Manager.
- **PagerDuty Subdomain** — The subdomain of your PagerDuty account.
 1. Log in to PagerDuty.
 2. Go to any URL to obtain the subdomain (the text between `https://` and `.pagerduty.com`). Example: `https://yourcompany.pagerduty.com` where `yourcompany` is the subdomain.



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

PagerDuty 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 PagerDuty 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 two default PagerDuty templates:

- pagerDutyCreateAndUpdate.xml
- pagerDutyCreateUpdateAndResolve.xml

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 PagerDuty Connector Instance:

```
emctl register_template connector
-t <template.xml>
-repos_pwd <repos password>
-cname "PagerDuty 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.

The screenshot shows the Oracle Enterprise Manager Cloud Control 13c interface. The breadcrumb trail is: Management Connectors > Configure Management Connector: PagerDuty Demo > Configure Management Connector: PagerDuty Demo. The 'Template' tab is selected. Below the tab, there is a table of templates used by this connector. The table has columns for Name, Type, and Description. The first row is selected.

Select	Name	Type	Description
<input checked="" type="radio"/>	Create Ticket Response	Inbound	This template handles responses from requests to create/update tickets in PagerDuty
<input type="radio"/>	Get Ticket Response	Inbound	This template handles responses from retrieving tickets in PagerDuty
<input type="radio"/>	PagerDuty Create and Update	Outbound	This template generates a request to create/update tickets in PagerDuty
<input type="radio"/>	PagerDuty Create, Update, and Resolve	Outbound	This template generates a request to create/update/resolve tickets in PagerDuty
<input type="radio"/>	Get Ticket Request	XML Outbound	This template generates a request to retrieve a ticket in PagerDuty

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.



Note:

If the template you delete has an incident rule associated with it, ticketing will not function for this particular incident rule after the deletion.

Editing Templates

You can edit a template directly using the Enterprise Manager console instead of using the EM CLI command line interface.

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 using the Enterprise Manager console.



Note:

You can also restore out-of-box templates that you have may edited or customized to their default values.

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 PagerDuty 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 PagerDuty, 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 PagerDuty connector.

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

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

ORACLE Enterprise Manager Cloud Control 13c

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: PagerDuty REST Ticketing Connector 13.5.0.0.0
 - Connector Name: PagerDuty Demo
 - Template Name:
 - PagerDuty Create and Update
 - PagerDuty Create, Update, and Resolve
- Send Notifications**
- Submit Corrective Action**
 Select a corrective action to be run when rule conditions are met. Only one corrective action is allowed.

Select corrective action

 - 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, for some events, 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

ORACLE Enterprise Manager Cloud Control 13c

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 ⓘ

▶ **Send Notifications**

▲ **Update Incident**
Automatically update the incidents to help triage your incoming incidents.

Assign to 🔍

Set priority to

Set status to

Escalate to

▲ **Create Ticket**

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

Connector Type

Connector Name

Template Name

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

✖ CPU Utilization is 100%, crossed warning (80) or critical (95) threshold.
Unassigned, Not acknowledged

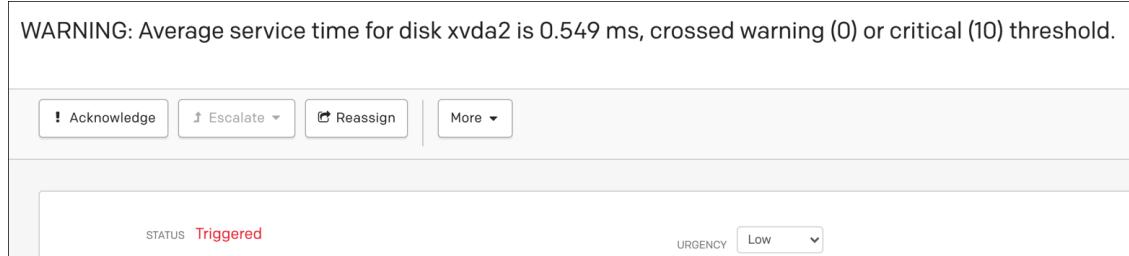
General Events Notifications My Oracle Support Knowledge **All Updates** Related Events Related Metrics

View ▾

Type	User	Message
D., Rule update		📘 Incident created by rule (Name = Metric events create incidents., Metrics Events Create Incidents; Owner = SYSMAN).

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 the PagerDuty console.



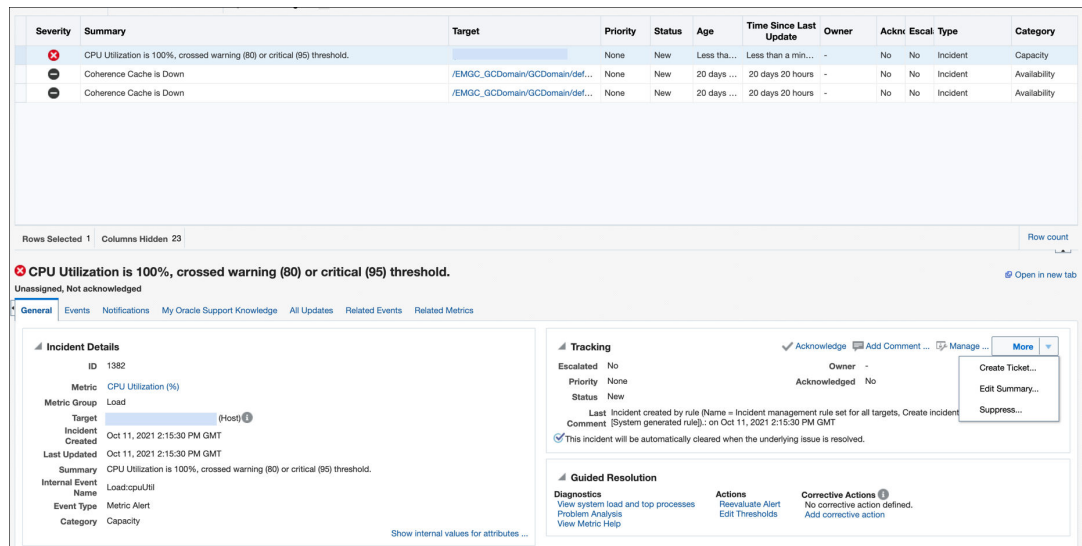
Manually Creating a Ticket

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.



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

Create Ticket [X]

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

Connector Type PagerDuty REST Ticketing Connector 13.5.0.0.0 ▾

Connector Name PagerDuty Demo ▾

Template Name ✓ PagerDuty Create and Update
PagerDuty Create, Update, and Resolve

OK Cancel

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 PagerDuty.

4

Using Default Templates

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

- Write incident information to `Description` (PagerDuty ticket description).
- Set the PagerDuty ticket summary based on the incident message.

The out-of-box templates are as follows:

- `pagerDutyCreateUpdateAndResolve.xml`
The `pagerDutyCreateUpdateAndResolve.xml` template sets the ticket status to *Resolved* when the event severity value becomes clear.
- `pagerDutyCreateAndUpdate.xml`
The `pagerDutyCreateAndUpdate.xml` template will update the ticket as details change in Oracle Enterprise Manager but does not close the ticket when the event severity becomes clear.

Reading Ticket Templates

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

Ticket creation mappings are the same for `pagerDutyCreateUpdateAndResolve.xml`, and `pagerDutyCreateAndUpdate.xml`.

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

PagerDuty Ticket Attributes	Enterprise Manager incident Attributes	Value
type	.	"incident" *

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

PagerDuty Ticket Attributes	Enterprise Manager incident Attributes	Value
service/id	\$\$SERVICE_ID\$	Set to the value of Service ID during configuration
service/type	.	"service_reference" *
body/type	.	"incident_body" *
body/details	Summary Severity Code Priority Notification Rule Name Notification Rule Owner Target Type Target Name Obj Owner Target Properties Last Updated Date Incident URL	Summary – The details about the incident that has fired Severity Code – The severity of the incident in Oracle Enterprise Manager Priority – The priority of the incident in Oracle Enterprise Manager Notification Rule Name – Name of the rule that created the ticket (Auto ticketing only) Notification Rule Owner – Name of the user that owns the rule that created the ticket (Auto ticketing only) Target Type – The type of target that the incident fired on Target Name – The name of the target that the incident fired on Obj Owner – Name of the user that owns the object the incident is fired on Target Properties – Additional names and values about the incident. Includes things like host, version, OS and platform Last Updated Date – Date the incident was created in Oracle Enterprise Manager Incident URL – A URL link that directs the user to the Oracle Enterprise Manager Incident
Title	Severity Code Summary	Severity Code – The severity of the incident in Oracle Enterprise Manager Summary – The details about the incident that has fired.
Urgency	Severity Code	The severity of the incident in Oracle Enterprise Manager. If the severity is Fatal or Critical, set Urgency to 'High' Otherwise, set the Urgency to 'Low'
Status	.	Hardcoded to 'resolve'.

 **Note:**

This is only in the
pagerDutyCreateUpdateAndResolve.xsl

Use the mapping table above as a reference to read the following XSLT file.

```
<?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 have two variables set:

- * Action: Used to figure out if the the connector is 'creating', 'updating' or 'closing' the PagerDuty Ticket.
- * Severity: Used to set the Urgency of the PagerDuty 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:when test="(emcf:SystemAttributes/emcf:SeverityCode = 'CLEAR')">CLOSE</
xsl:when>
    <xsl:otherwise>UPDATE</xsl:otherwise>
  </xsl:choose>
</xsl:variable>
<xsl:variable name="severity">
  <xsl:choose>
    <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'FATAL'">high</
xsl:when>
    <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'CRITICAL'">high</
xsl:when>
    <xsl:otherwise>low</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
PagerDuty.
  -->
  <Method>
    <xsl:choose>
      <xsl:when test="$action = 'CREATE'">POST</xsl:when>
      <xsl:when test="$action = 'UPDATE'">PUT</xsl:when>
      <xsl:when test="$action = 'CLOSE'">PUT</xsl:when>
      <xsl:otherwise>ERROR</xsl:otherwise>
    </xsl:choose>
  </Method>
  <Variables>
    <Variable>
      <Name>TKT_ID</Name>
      <Value>
        <xsl:value-of select="normalize-space(emcf:TicketID)" />
      </Value>
    </Variable>
  </Variables>
  <Headers>
    <Header>
      <Name>Authorization</Name>
      <Value>Token token=$TOKEN$</Value>
    </Header>
    <Header>
      <Name>Accept</Name>
      <Value>application/vnd.pagerduty+json;version=2</Value>
    </Header>
    <Header>
      <Name>From</Name>
      <Value>$USER_EMAIL$</Value>
    </Header>
  </Headers>
  <!--
    The body is used to send data to PagerDuty. By default the connector sends data
```

either On Create, Update or Close.

Fields that are sent when the ticket is created are found under the following header: `<xsl:if test="$action = 'CREATE'">`
 * Details: This fills in the 'details' section of the ticket in PagerDuty.

Fields that are sent when the ticket is created or updated are found under the following header: `<xsl:if test="$action = 'CREATE' or $action = 'UPDATE'">`
 * Title: This is the title of the ticket in PagerDuty.
 * Urgency: This sets the urgency ('High' or 'Low') based on the incident's severity.

Fields that are sent when the ticket is closed are found under the following header: `<xsl:if test="$action = 'CLOSE'">`
 * Status: This is hard-coded to 'resolved' to close the ticket in PagerDuty.

```
-->
<Body>
  <object>
    <object name="incident">
      <string name="type">incident</string>
      <object name="service">
        <string name="id">${SERVICE_ID}</string>
        <string name="type">service_reference</string>
      </object>
      <xsl:if test="$action = 'CREATE'">
        <object name="body">
          <string name="type">incident_body</string>
          <string name="details">Incident created in 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 Enterprise Manager, click here: <xsl:value-of
select="emcf:SystemAttributes/emcf:IncidentURL" />
  </string>
</object>
</xsl:if>
```

```

    <xsl:if test="$action = 'CREATE' or $action = 'UPDATE'">
      <string name="title">
        <xsl:value-of select="emcf:SystemAttributes/emcf:SeverityCode" />:
<xsl:value-of select="emcf:SystemAttributes/emcf:Summary" />
      </string>
      <string name="urgency">
        <xsl:value-of select="$severity" />
      </string>
    </xsl:if>
    <xsl:if test="$action = 'CLOSE'">
      <string name="status">resolved</string>
    </xsl:if>
  </object>
</object>
</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 Make Warning Severity High Urgency

To create a template to change warning severity to high urgency in PagerDuty, add the following attribute in the template under the variable severity:

```
<xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'WARNING'">high</xsl:when>
```

Example 4-2 Altering the Title Name

If you only want the incident summary to appear as a PagerDuty ticket title instead of both message and severity, modify the following attribute:

```

<string name="title">
  <xsl:value-of select="emcf:SystemAttributes/emcf:Summary" />
</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 HPD:IncidentInterface_Create,HPD:IncidentInterface 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 PagerDuty:

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>Database1</code> or <code>yourhost.example.com</code> .
TargetProperties	<p>TargetProperties store environmental or usage context information specific to the target. For example, for Host Target, the name /value pair of TargetProperties are:</p> <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 <p>These are out-of-box user-defined target properties. If additional target properties are added, they are displayed with ticket information.</p>
Summary	Description of the incident. For example: CPU Utilization is 100%, crossed warning (80) or critical (95) threshold.
Severity	Severity of the incident: <code>critical</code> , <code>warning</code> , <code>clear</code> , or <code>down</code> .
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 PagerDuty forms, adhere to the following format:

Example 4-3 Template Format for Custom PagerDuty 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'">PUT</xsl:when>
          <xsl:when test="$action = 'CLOSE'">PUT</xsl:when>
          <xsl:otherwise>ERROR</xsl:otherwise>
        </xsl:choose>
      </Method>
      <Variables>
        <Variable>
          <Name>TKT_ID</Name>
          <Value>
            <xsl:value-of select="normalize-space(emcf:TicketID)" />
          </Value>
        </Variable>
      </Variables>
      <Headers>
        <Header>
          <Name>Authorization</Name>
          <Value>Token token=$TOKEN$</Value>
        </Header>
        <Header>
          <Name>Accept</Name>
          <Value>application/vnd.pagerduty+json;version=2</Value>
        </Header>
        <Header>
          <Name>From</Name>
          <Value>$USER_EMAIL$</Value>
        </Header>
      </Headers>
      <Body>
        * [Insert body changes (details you are sending to PagerDuty) here]
      </Body>
    </Message>
  </xsl:template>
</xsl:stylesheet>
```


Index

C

- creating
 - trouble tickets manually (PagerDuty), [3-4](#)
- customizing
 - PagerDuty ticket templates, [4-5](#)

D

- default templates, using for PagerDuty Connector, [4-1](#)
- defining
 - PagerDuty ticket templates, [4-6](#)

F

- format for creating
 - PagerDuty ticket templates, [4-6](#)

P

- PagerDuty Connector
 - customizing ticket templates, [4-5](#)

PagerDuty Connector (*continued*)

- default templates, using, [4-1](#)
- defining new ticket templates, [4-6](#)
- format for creating ticket templates, [4-6](#)
- replacing a template, [2-11](#)
- ticket templates, format for creating, [4-6](#)

R

- replacing a template
 - PagerDuty Connector, [2-11](#)

T

- templates
 - customizing PagerDuty ticket templates, [4-5](#)
 - defining new PagerDuty ticket templates, [4-6](#)
 - replacing, [2-11](#)
- ticket templates
 - customizing PagerDuty ticket templates, [4-5](#)
 - defining new PagerDuty ticket templates, [4-6](#)
- trouble tickets
 - creating manually for PagerDuty Connector, [3-4](#)