

Oracle® Enterprise Manager

Microsoft Systems Center Operations Manager (SCOM) Connector Installation and Configuration Guide



13c Release 5
F37151-03
December 2025

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

ORACLE®

F37151-03

Copyright © 2013, 2025, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	i
Documentation Accessibility	i
Conventions	i

1 Microsoft SCOM REST Event Connector Overview

Terminology	2
Supported Environments and Versions	2
Prerequisite Installations and Configurations	3

2 Installing the Microsoft SCOM REST Event Connector and Web Service

Downloading the Connector	1
Downloading the Installation Files	1
Importing the Management Pack	2

3 Configuring the Microsoft SCOM REST Event Connector and Web Service

Configuring the Event Connector Web Service	1
Running the Event Connector Web Service	3
Running the Event Connector Web Service in the Background	3
Configuring the Oracle Enterprise Manager Management Connector	5

4 Viewing Events in the Microsoft SCOM Operations Console

5 Enabling Secure Communication

Importing the Server Certificate in Oracle Enterprise Manager	2
---------------------------------------------------------------	---

6 Customization of the Request Templates

7 Migrating From SCOM Connector 13.2.2.0.0 or Older Version

Moving Custom Logic from Previous to New Templates	1
Forwarding Events to the New Connector	3
Uninstalling the Alert Creator Management Pack	4
Uninstalling the Oracle SCOM Agent	4
Uninstalling the Microsoft SCOM Web Service on UNIX	4
Uninstalling the Microsoft SCOM Web Service on Windows	4
Uninstalling the Microsoft SCOM Event Connector	5

Preface

This document provides the required information to install and configure the Microsoft System Center Operations Manager (SCOM) REST Event Connector that integrates Oracle Enterprise Manager with SCOM management tools and help desk systems.

Audience

This guide is written for Oracle Enterprise Manager system administrators who want to install and configure the Microsoft SCOM REST Event Connector to enable integration between Oracle Enterprise Manager and Microsoft SCOM.

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 that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/us/corporate/accessibility/support/index.html#info> or visit <http://www.oracle.com/us/corporate/accessibility/support/index.html#trs> if you are hearing impaired.

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

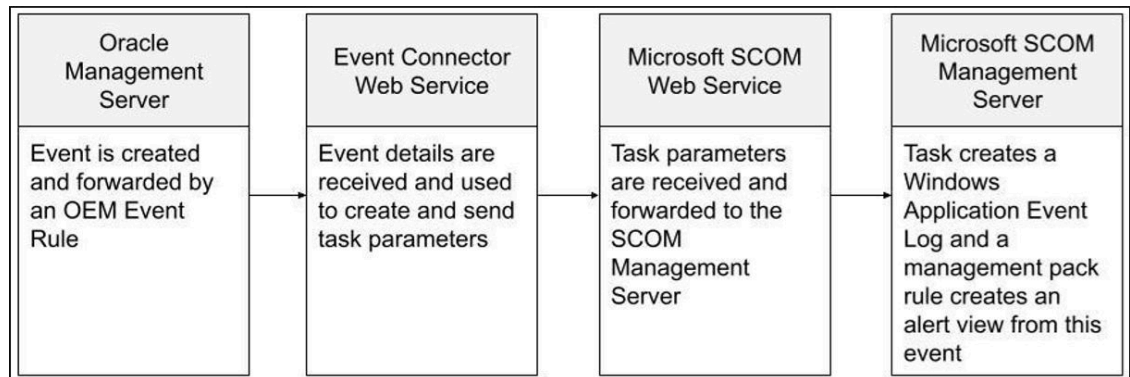
Microsoft SCOM REST Event Connector Overview

The SCOM REST Event Connector sends newly created events from Oracle Enterprise Manager to a Microsoft SCOM Management Server and annotates event updates.

The connector supports the following features:

- Sharing of event information from Oracle Enterprise Manager to the Microsoft SCOM Management Server
- Customization of event to alert mappings between Oracle Enterprise Manager and the Microsoft SCOM Management Server
- Sending of event updates via annotations

The connector does not communicate directly with the Microsoft SCOM Management Server. Instead, it relies on two web services that reside between the OMS and the Microsoft SCOM Management Server. When an event is created or updated in Oracle Enterprise Manager, the connector sends the event's details to the Event Connector Web Service via a REST call. The Event Connector Web Service then converts this data into task parameters and sends them via a POST call to the `/OperationsManager/data/submitTask` endpoint of the Microsoft SCOM Web Service. Once the task parameters are received by the Microsoft SCOM Management Server, a local task is executed to create a Windows Application Event Log entry. Finally, a management pack rule creates an alert view in the Microsoft SCOM Operations console based on the log entry. The following diagram visualizes this communication. Note that this diagram illustrates the flow of communication between services and is not intended to imply any specific network topology requirements. Any or all of these services can run on the same server, depending on your specific topology requirements.



Both sides of communication on the Event Connector Web Service use the following:

- REST-based requests
- Basic authentication
- Optional secure communication via SSL/TLS

The following details are sent from Oracle Enterprise Manager:

- Resolution State
- Event ID

- Target Host
- Target Type
- Target Name
- Event URL
- Severity
- Priority
- Description of the event
- Creation/update timestamp

Topics

- [Terminology](#)
- [Supported Environments and Versions](#)
- [Prerequisite Installations and Configurations](#)

Terminology

- **ECWS_HOME:** The root installation directory, where the `WebService.jar` file is placed.

Note

`ECWS_HOME` does not refer to an assumed environment variable and is only used for convenience within the context of this document.

- **JAVA_HOME:** The home directory that points to Java 17 or later.

Note

`JAVA_HOME` does not refer to an assumed environment variable and is only used for convenience within the context of this document.

- **Event Connector Web Service:** The web service that belongs to the Oracle Enterprise Manager Event Connector. This resides between the OMS and the Microsoft SCOM Web Service. It provides the endpoint that is defined within Oracle Enterprise Manager from the connector configuration page.
- **Microsoft SCOM Web Service:** An optional installation feature when installing the Microsoft SCOM Management Server. This Internet Information Services (IIS) site provides the management API endpoint, `/OperationsManager/data/submitTask`, which is required by the Event Connector Web Service.

Supported Environments and Versions

- 64-bit Windows or 64-bit Linux (where Event Connector Web Service is running)
 - Minimum Requirements: 4 CPU Cores, 2.5 GHz
 - Available Disk Space: 130 MB (to account for log growth)
- Oracle Enterprise Manager 13.5 and 24.1 and later

- Microsoft System Center Operations Manager 2019 or later (on a local or remote host to the Event Connector Web Service)

Prerequisite Installations and Configurations

- Microsoft SCOM Operations console
- Microsoft SCOM Web Service, configured and running. For information, see [Install the Operations Manager web console](#) in Microsoft documentation.
- Java 17 or later on the host running the Event Connector Web Service
- Access to Oracle Enterprise Manager and the software library

2

Installing the Microsoft SCOM REST Event Connector and Web Service

Microsoft SCOM's API uses a model that prevents Oracle Enterprise Manager from sending alerts directly through the Event Connector framework. To work around this limitation, this connector includes a web service and a management pack, which enables the communication of alerts from Oracle Enterprise Manager to SCOM.

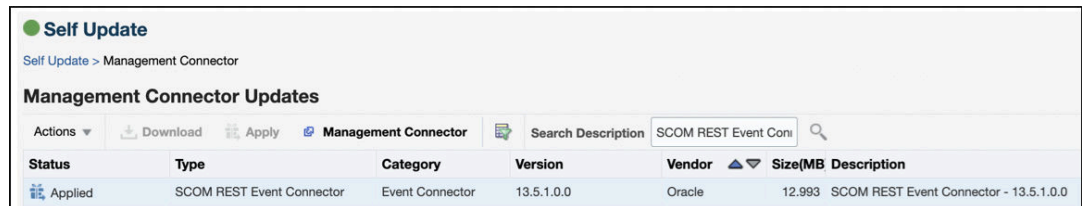
In addition to configuring the connector, you must install the web service and the management pack. The installation files for both are located within the **Self Update** download for this connector.

Topics

- [Downloading the Connector](#)
- [Downloading the Installation Files](#)
- [Importing the Management Pack](#)

Downloading the Connector

1. From the **Setup** menu, select **Extensibility**, then select **Self Update**.
2. On the **Self Update** page, select **Management Connector**.
3. Enter **SCOM REST Event Connector** in the **Search Description** field and select the latest version from the results.
4. Click **Download** to download the connector and then select the connector and click **Apply**.



5. From the **Actions** menu, select **Export to** and export a .zip file of the connector files to a desired location.

Downloading the Installation Files

1. Unzip the folder containing the connector files and open the `archives/` directory.
2. Transfer the following files to appropriate locations:
 - `OracleEnterpriseManager.Alert.Creator.xml`: Transfer to the same host where Microsoft SCOM is installed.`OracleEnterpriseManager.Alert.Creator.xml` or the Oracle Enterprise Manager Alert Creator management pack creates the monitoring views and the tasks required to generate alerts in the Microsoft SCOM Operations console. Make a note of the path to

`OracleEnterpriseManager.Alert.Creator.xml` on the host, as this will be required when importing it. For information, see [Importing the Management Pack](#).

- `WebService.jar`: Transfer to the host where you want the Event Connector Web Service to run. Ensure that this host can communicate with both the Microsoft SCOM Web Service and the OMS environment.

Importing the Management Pack

1. In the **Administration** pane of the Microsoft SCOM Operations console, select **Administration**, and then select **Management Packs**.
2. Right-click on **Management Packs** and select **Import Management Packs**.
The **Import Management Packs** dialog box is displayed.
3. Select **Add**, and then select **Add from disk**.
If prompted during the process, select **Yes** to search for missing local dependencies.
4. Navigate to the directory where the `OracleEnterpriseManager.Alert.Creator.xml` (Oracle Enterprise Manager Alert Creator management pack) file is located, and select the file to import.
5. Verify that the Oracle Enterprise Manager Alert Creator management pack is successfully imported by checking the following in the Microsoft SCOM Operations console:
 - The **Oracle Enterprise Manager Alert Creator** management pack is displayed in the **Administration** view.
 - The **Oracle Enterprise Manager Event Monitoring** alerts folder is displayed in the **Monitoring** view.
 - The **Create Windows Event from Oracle Enterprise Manager Event** management pack object task is displayed in the **Authoring** view.

For more information, see [Importing a Management Pack](#) in Microsoft documentation.

3

Configuring the Microsoft SCOM REST Event Connector and Web Service

The Event Connector Web Service is the API that receives the request directly from Oracle Enterprise Manager and performs the follow up actions of authenticating to the SCOM server and translating the alert into a format that the management pack can interpret.

Topics

- [Configuring the Event Connector Web Service](#)
- [Running the Event Connector Web Service](#)
- [Running the Event Connector Web Service in the Background](#)
- [Configuring the Oracle Enterprise Manager Management Connector](#)

Configuring the Event Connector Web Service

1. Log in to the host where you will run the Event Connector Web Service.

This can be a Windows or Linux host, provided it has Java 17 or later available.

2. Open a terminal window and navigate to the directory containing `WebService.jar`.

This directory will be referred to as `ECWS_HOME` in this chapter.

After configuring the Event Connector Web Service, two additional folders, `logs/` and `config/`, are created inside it. It is recommended that you choose an empty folder that can be dedicated solely to the Event Connector Web Service files.

3. Execute the following command, where `JAVA_HOME` is set to Java 17 or later:

```
$JAVA_HOME/bin/java -jar WebService.jar -wsconfig
```

4. Select **Configure connector webservice** and provide the following inputs:

- **Webservice Username** and **Password**: The credentials used to authenticate Oracle Enterprise Manager requests to the Event Connector Web Service. These credentials are also required the next time you run `WebService.jar` to view or modify existing configurations. Note that these credentials are not the Microsoft SCOM Web Service credentials.
- **Webservice Port**: The port on which the Event Connector Web Service listens. If the Event Connector Web Service is on a different host than the Oracle Management Service, you must ensure that the port is accessible from the host where the Oracle Management Service is running. The default port is `8080`.
- **SCOM Username** and **Password**: The credentials that allow the Event Connector Web Service to authenticate to the Microsoft SCOM Web Service. Do *not* include the domain name when providing the **SCOM Username**.
- **SCOM Windows Domain**: The domain to which the specified SCOM user belongs.

- **SCOM Port:** The port on which the Microsoft SCOM Web Service listens. If the SCOM Web Service is on a different host than the Event Connector Web Service, you must ensure that the port is accessible from the host where the Event Connector Web Service is running. The default port is 80.
 - **SCOM Server Hostname:** The resolvable host name where the Microsoft SCOM Web Service is running.
 - **SCOM Management Server:** The name that SCOM uses to identify its management server.
 - **Windows Event ID:** The integer value that identifies an incoming event from Oracle Enterprise Manager. Select a value that does not conflict with other Event IDs in the Windows Application Logs. If no ID is provided, the configuration will default to 900.
 - **Enable Web Service SSL:** Enter **yes** or **no**. This enables one-way SSL between the OMS and the Event Connector Web Service. You will also be prompted to provide the keystore password. For more information, see [Enabling Secure Communication](#).
 - **Enable SCOM SSL:** Enter **yes** or **no**. Enable this option when the Microsoft SCOM Web Service is configured for secure communication. This allows the Event Connector Web Service to trust the certificate presented by the Microsoft SCOM Web Service. You will also be prompted to provide the truststore password. For more information, see [Enabling Secure Communication](#).
5. On the main screen, select **View current configurations** to verify that the information provided is correct.

Here's an example of the configuration:

```
ORACLE ENTERPRISE MANAGER SCOM CONNECTOR CONFIGURATION MANAGER
===== Current Configurations =====

Webservice Username=ConnectorWebserviceUser
Webservice Password=*****
Webservice Port=8443
SCOM Username=ScomWebserviceUser
SCOM Windows Domain=CONTOSO
SCOM Password=*****
SCOM Port=443
SCOM Server Hostname=scom01.example.com
SCOM Management Server=SCOM01
Windows Event ID=900
Enable Web Service SSL=yes
Enable SCOM SSL=yes
Key Store Password=*****
Trust Store Password=*****

[Enter] to continue..
```

If any of the information is incorrect, select **Update current configurations** on the main screen. After verifying that the configuration details are correct, select **Exit**. You can modify the configuration at any time by executing the `WebService.jar` file with the `-wsconfig` flag. Note that if you forget or misplace the credentials created for the Event Connector Web Service, delete the configuration file at `<ECWS_HOME>/config/WebServiceConfig.json` and reconfigure the connector.

Running the Event Connector Web Service

To run the service, navigate to `ECWS_HOME` and execute the `WebService.jar` file as mentioned in the configuration steps, but omit the `-wsconfig` flag:

```
$JAVA_HOME/bin/java -jar WebService.jar
```

If successful, the following message is displayed in the terminal session:

```
Connector webservice successfully started on port <PORT>
```

If no message is displayed or the application exits unexpectedly, re-run the command in debug mode using the `-logLevel` flag. This generates more detailed messages in the `ecws.log` files, located in the `<ECWS_HOME>/logs` directory. Here's an example command:

```
$JAVA_HOME/bin/java -jar WebService.jar -logLevel=DEBUG
```

You can also search the logs for the following line, which will be followed by its cause:

```
Unable to start web service due to the following
```

Running the Event Connector Web Service in the Background

By default, the Event Connector Web Service runs in a command or terminal window that must remain open. However, it can be manually configured on Windows and Linux to run as a background process at startup. The required steps depend on your Operating System (OS), so refer to the relevant OS documentation. However, here are example steps that work in most environments.

Windows Example

1. Open a text editor and paste the following script after replacing `<ECWS_HOME>` and `<JAVA_HOME>` with the appropriate paths:

```
@echo off
cd /d "%ECWS_HOME%"
"%JAVA_HOME%\bin\java.exe" -jar "WebService.jar" >> "logs\ecws.log" 2>&1
```

2. Save the script as a batch file named `EcwsStartScript.bat`.
3. Open Windows Task Scheduler.
4. From the **Actions** menu, select **Create Task**. The **Create Task** dialog box is displayed.
5. In the **General** tab:
 - a. Enter `ecws` in the **Name** field.
 - b. Select the **Run whether user is logged on or not** option.
 - c. Select the **Run with highest privileges** option.
6. In the **Triggers** tab:

- a. Click **New**.
The **New Trigger** dialog box is displayed.
 - b. From the **Begin the task** menu, select **At startup**.
 - c. Ensure that the **Enabled** option is selected.
 - d. Click **OK**.
7. In the **Actions** tab:
- a. Click **New**.
The **New Action** dialog box is displayed.
 - b. From the **Action** menu, select **Start a program**.
 - c. Set the **Program/script** field to `<ECWS_HOME>\EcwsStartScript.bat`.
 - d. Click **OK**.
8. In the **Conditions** tab, ensure that the **Start the task only if the computer is on AC power** option is *not* selected.
9. In the **Settings** tab, ensure that the **Stop the task if it runs longer than...** option is *not* selected.
10. Click **OK**.
11. Open a command prompt as Administrator and use the following commands to:
- Start the task:

```
schtasks /run /tn "ecws"
```
 - Stop the task:

```
schtasks /end /tn "ecws"
```

Linux Example

1. Create a new service file at `/etc/systemd/system/ecws.service`.
2. Replace `<ECWS_HOME>` with the appropriate installation path, for example, `/opt/ecws`, and copy the following contents into `ecws.service`:

```
[Unit]
Description=OEM to SCOM Event Connector Web Service
After=network.target

[Service]
WorkingDirectory=<ECWS_HOME>
ExecStart=/usr/bin/java -jar <ECWS_HOME>/WebService.jar
Restart=always
RestartSec=15
SuccessExitStatus=143

[Install]
WantedBy=multi-user.target
```

3. Execute the following commands to:

- Load the service:

```
sudo systemctl daemon-reload
```
- Enable the service:

```
sudo systemctl enable --now ecws
```
- 4. The service will be initiated at system startup but can be controlled using the following commands:
 - Start the service:

```
sudo systemctl start ecws
```
 - Stop the service:

```
sudo systemctl stop ecws
```

Configuring the Oracle Enterprise Manager Management Connector

1. Ensure that the Event Connector Web Service is configured and running.
2. Confirm that the OMS host can access the Event Connector Web Service by performing the following steps:
 - a. Open the `test-connection` endpoint in a browser. For example, if the Event Connector Web Service is running on host `mycompany.com`, port 8443, and secure communication is enabled, the URL would be:

```
https://mycompany.com:8443/test-connection
```

You will receive an authentication prompt.

- b. Enter the credentials you created for authenticating the Event Connector Web Service. A `json` response indicating that the test was successful is displayed. If you do not see the message, resolve the communication or configuration issue before proceeding.
3. From the **Setup** menu, select **Extensibility**, then select **Management Connectors**, and create a new **SCOM REST Event Connector** instance.
4. On the configuration page, ensure that at least the following web service details are entered or updated:
 - In the **createEvent** and **updateEvent** fields, replace `<hostname>` with the resolvable host name that your Event Connector Web Service is running on.
 - In the **createEvent** and **updateEvent** fields, update the port number if you are not using the default 8080.
 - In the **createEvent** and **updateEvent** fields, replace `http` with `https` if secure communication is enabled.
 - Update the **Webservice Username** and **Webservice Password** with the credentials created for authenticating to the Event Connector Web Service.

- Configure proxy settings only if using a proxy (optional).

Note

This configuration page information applies only to communication between Oracle Enterprise Manager and the Event Connector Web Service. These fields do not apply to communication between the Event Connector Web Service and the Microsoft SCOM Web Service.

Here's an example that shows the minimal configuration requirements:

* Web Service End Points	Operation	Web Service End Point (URL)
	createEvent	<input type="text" value="http://example.com:8080/submit-task"/>
	updateEvent	<input type="text" value="http://example.com:8080/submit-task"/>

* Webservice Username

* Webservice Password

5. Click **OK** to save the configuration.

4

Viewing Events in the Microsoft SCOM Operations Console

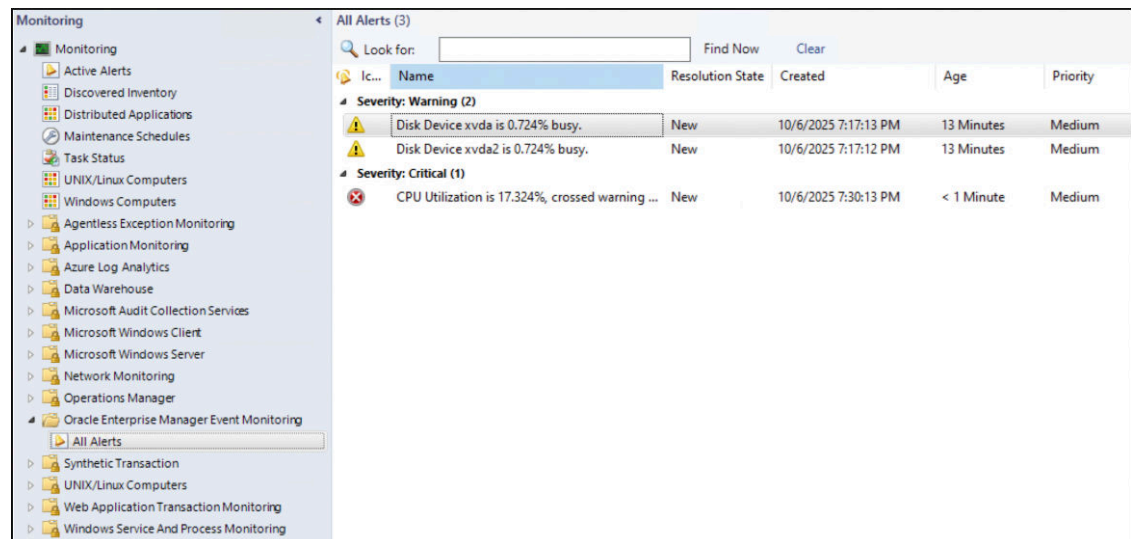
After an event is sent by an Incident Rule in Oracle Enterprise Manager, you can view it in the Microsoft SCOM Operations console.

For information on how to send events, see *Sending Events to an Event Connector in Oracle Enterprise Manager Monitoring Guide*.

Note

The content in the Sending Events to an Event Connector section uses IBM Tivoli Netcool/OMNIBus as an example, however, it describes the generic process for sending events to any Event Connector.

After an event has been sent to SCOM, you can view it in the Microsoft SCOM Operations console. To do so, select **Monitoring**, **Oracle Enterprise Manager Event Monitoring**, and then **All Alerts**.



Several custom fields are associated with each incident. There is a limitation that prevents the management pack from renaming these fields, and the following mapping defines the meaning of each custom field:

- Custom Field 1: Target Name
- Custom Field 2: Target Type
- Custom Field 3: Target Host
- Custom Field 4: Create Date
- Custom Field 5: Last Update
- Custom Field 6: URL

- Custom Field 7: Owner
- Custom Field 8: Oracle Enterprise Manager Event ID
- Custom Field 9: Message
- Custom Field 10: Updates Annotated

Note

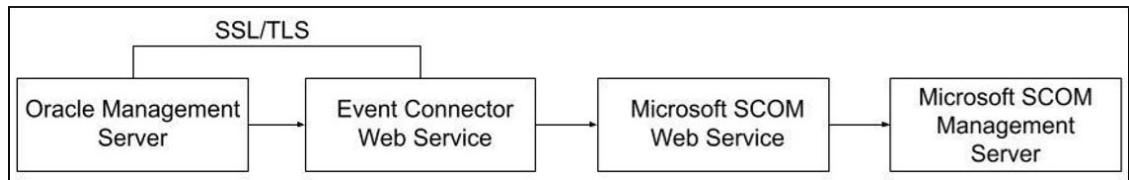
The Microsoft SCOM framework does not allow the modification of an alert's severity once the alert has been created. Because of this, any subsequent severity changes are recorded in Custom Field 10.

5

Enabling Secure Communication

By default, the Event Connector Web Service uses insecure communication. You can configure secure communication between the OMS and the Event Connector Web Service, the Event Connector Web Service and the Microsoft SCOM Web Service, or both. The appropriate setup for your environment depends on your network topology.

Configuration Scenario: Enabling SSL/TLS Between the OMS and the Event Connector Web Service



1. Navigate to `<ECWS_HOME>/config/`.
2. Create a keystore with a valid server certificate to be presented by the Event Connector Web Service. The keystore must include the following attributes:
 - Type: PKCS12
 - Filename: `OemScomKeystore.p12`
 - Keystore location: `<ECWS_HOME>/config/` directory
 - Certificate alias: `oemscmconnector`

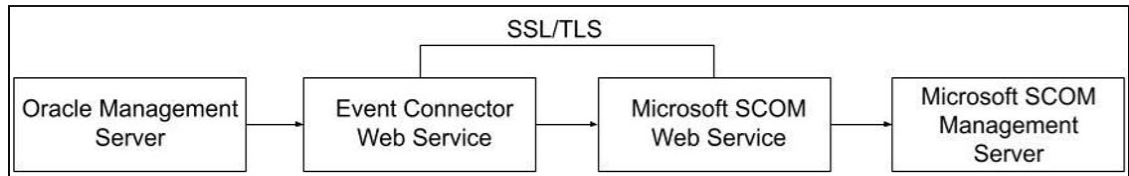
Note

If any of the above attributes are missing or incorrect when secure communication is expected, the web service will fail to start and a message will be generated to `ecws.log`.

Although the approach for obtaining this may vary, the following is an example command:

```
%JAVA_HOME%\bin\keytool" -importkeystore ^
-srckeystore scom_server.pfx
-srcstoretype PKCS12
-srcstorepass "<password>"
-srcalias "scomserver"
-destkeystore OemScomKeystore.p12
-deststoretype PKCS12
-deststorepass "<password>"
-destalias oemscmconnector
```

3. Import the certificate into the OMS truststore. For information, see [Importing the Server Certificate in Oracle Enterprise Manager](#).

Configuration Scenario: Enabling SSL/TLS Between the Event Connector Web Service and the Microsoft SCOM Web Service

1. Navigate to the `config/` directory that was populated when first executing `WebService.jar`.
2. Create a truststore and import the valid server certificate from the Microsoft SCOM Web Service. The truststore must include the following attributes:
 - Type: PKCS12
 - Filename: `OemScomTruststore.p12`
 - Truststore location: `<ECWS_HOME>/config/` directory

Note

If any of the above attributes are missing or incorrect when secure communication is expected, the web service will fail to start and a message will be generated to `ecws.log`.

Although the approach for obtaining this may vary, the following is an example command:

```
keytool -importcert
  -alias scomserver
  -file scom_server.crt
  -keystore OemScomTruststore.p12
  -storetype PKCS12
  -storepass <password>
```

Importing the Server Certificate in Oracle Enterprise Manager

1. Obtain the certificate authority and any necessary intermediate certificates in Base64 format.
2. Log in to your OMS host and edit the following file after creating a backup: `$INSTANCE_HOME/sysman/config/b64LocalCertificate.txt`
3. Append the contents of your certificate to the end of the contents of the `b64LocalCertificate.txt` file. Do not include blank lines, comments, or any other special characters. Each new certificate in this file should only contain the following format:

```
-----BEGIN CERTIFICATE-----
<Certificate contents in Base64 format>
-----END CERTIFICATE-----
```

- Restart OMS by running the following commands:

```
emctl stop oms  
emctl start oms
```

Note

Do not run the `emctl secure oms/agent` command after adding the external certificate to the `b64LocalCertificate.txt` file. If you run the `emctl secure` command later, then repeat these steps to make sure the external certificate exists in the `b64LocalCertificate.txt` file.

6

Customization of the Request Templates

You can modify the `request.xml` templates if the default template does not conform to your requirements:

- `createEvent_request.xml`
- `updateEvent_request.xml`

The `response.xml` templates must *not* be modified:

- `createEvent_response.xml`
- `updateEvent_response.xml`

Note that attribute names should not be altered, added, or removed in the XSL template. Attribute names use the `<string name="attributeName">` format. The content within these attributes may be modified, except for the following attributes, which serve backend purposes and must not be changed:

- `requestType`
- `id`
- `resolutionState`

Note

Custom attributes are not supported. If custom content is required, place it within the content of the `message` attribute.

The most commonly modified attributes are those where the default mappings might not conform with internal alerting practices. These include:

- `severityLevel`
- `severityCode`
- `priorityCode`

Only the `createEvent_request.xml` template contains the `severityLevel` attribute. This attribute must always match the corresponding `severityCode`, as defined by the following mapping rules:

<code>severityLevel</code>	<code>severityCode</code>
Information	0
Warning	1
Error	2

The following example shows how to modify the `createEvent_request.xml` template to map a severity of `Warning` in Oracle Enterprise Manager to an `Error` alert in SCOM, and an Oracle Enterprise Manager priority of `Medium` to a priority code of 2 in SCOM:

createEvent_request.xml (Default)

```

<string name="severityLevel">
  <xsl:choose>
    <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode =
'MINOR_WARNING' ">Warning</xsl:when>
    <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode =
'WARNING' ">Warning</xsl:when>
    <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode =
'CRITICAL' ">Error</xsl:when>
    <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'FATAL' ">Error</
xsl:when>
    <xsl:otherwise>Information</xsl:otherwise>
  </xsl:choose>
</string>

<!-- ... -->

<string name="severityCode">
  <xsl:choose>
    <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode =
'MINOR_WARNING' ">1</xsl:when>
    <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'WARNING' ">1</
xsl:when>
    <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'CRITICAL' ">2</
xsl:when>
    <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'FATAL' ">2</
xsl:when>
    <xsl:otherwise>0</xsl:otherwise>
  </xsl:choose>
</string>

<string name="priorityCode">
  <xsl:choose>
    <xsl:when test="emcf:SystemAttributes/emcf:Priority = 'Low' ">0</xsl:when>
    <xsl:when test="emcf:SystemAttributes/emcf:Priority = 'Medium' ">1</
xsl:when>
    <xsl:when test="emcf:SystemAttributes/emcf:Priority = 'High' ">2</xsl:when>
    <xsl:when test="emcf:SystemAttributes/emcf:Priority = 'Very High' ">2</
xsl:when>
    <xsl:when test="emcf:SystemAttributes/emcf:Priority = 'Urgent' ">2</
xsl:when>
    <xsl:otherwise>1</xsl:otherwise>
  </xsl:choose>
</string>

```

createEvent_request.xml (Modified with changes in **bold**)

```

<string name="severityLevel">
  <xsl:choose>
    <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode =
'MINOR_WARNING' ">Warning</xsl:when>

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

```

```

        <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode =
'CRITICAL' ">Error</xsl:when>
        <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'FATAL' ">Error</
xsl:when>

        <xsl:otherwise>Information</xsl:otherwise>
    </xsl:choose>
</string>

<!-- ... -->

<string name="severityCode">
    <xsl:choose>
        <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode =
'MINOR_WARNING' ">1</xsl:when>

        <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'WARNING' ">2</
xsl:when>
        <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'CRITICAL' ">2</
xsl:when>
        <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'FATAL' ">2</
xsl:when>

        <xsl:otherwise>0</xsl:otherwise>
    </xsl:choose>
</string>

<string name="priorityCode">
    <xsl:choose>
        <xsl:when test="emcf:SystemAttributes/emcf:Priority = 'Low' ">0</xsl:when>

        <xsl:when test="emcf:SystemAttributes/emcf:Priority = 'Medium' ">2</
xsl:when>
        <xsl:when test="emcf:SystemAttributes/emcf:Priority = 'High' ">2</xsl:when>
        <xsl:when test="emcf:SystemAttributes/emcf:Priority = 'Very High' ">2</
xsl:when>
        <xsl:when test="emcf:SystemAttributes/emcf:Priority = 'Urgent' ">2</
xsl:when>

        <xsl:otherwise>1</xsl:otherwise>
    </xsl:choose>
</string>

```

7

Migrating From SCOM Connector 13.2.2.0.0 or Older Version

If you are using a previous SCOM connector, note that version 13.5.1.0.0 uses a different backend logic. Before installing and configuring the new connector, you must completely uninstall the previous connector and its iWave web service.

If you have previously customized any of your request templates, retain a copy for reference before removing the connector. You will need to migrate your customizations from the previous template to the new template, because the older format is incompatible with the new REST version of the connector.

Note

Uninstalling the previous SCOM connector will remove all the current alerts generated by the connector service, regardless of their status. Before uninstalling, review the currently open alerts or export all open alerts using Microsoft documentation. Due to limitations within SCOM, these cannot be reimported to the new connector's alert table, but can be maintained for historical accounting.

Topics

- [Moving Custom Logic from Previous to New Templates](#)
- [Forwarding Events to the New Connector](#)
- [Uninstalling the Alert Creator Management Pack](#)
- [Uninstalling the Oracle SCOM Agent](#)
- [Uninstalling the Microsoft SCOM Web Service on UNIX](#)
- [Uninstalling the Microsoft SCOM Web Service on Windows](#)
- [Uninstalling the Microsoft SCOM Event Connector](#)

Moving Custom Logic from Previous to New Templates

Here is information on how to move customization from previous request templates to new templates.

If you have customized your previous template, the most common areas will be within the <severity>, <priority>, or <extended-fields> tags. Moving customized logic to the new templates will require planning, since you cannot copy and paste the previous template.

severity Tags in the Previous and New Templates

The new template includes `severityLevel` and `severityCode`. These tags replace `severity` in the previous template. Both `severityLevel` and `severityCode` represent the same concept and should be mapped in a 1:1 relationship according to the following logic:

Previous Template	New Template
severity=Information	severityLevel=Information and severityCode=0
severity=Warning	severityLevel=Warning and severityCode=1
severity=Error	severityLevel=Error and severityCode=2

For example, if you had the following warning mapping inside the severity tags of your previous template:

```
<severity>
  <xsl:choose>
    ...
    <xsl:when test="a:SystemAttributes/a:SeverityCode = 'WARNING'">Warning</
xsl:when>
    ...
  </xsl:choose>
</severity>
```

The new template should contain the following two mappings:

```
<string name="severityLevel">
  <xsl:choose>
    ...
    <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode =
'WARNING'">Warning</xsl:when>
    ...
  </xsl:choose>
</string>

...

<string name="severityCode">
  <xsl:choose>
    ...
    <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'WARNING'">1</
xsl:when>
    ...
  </xsl:choose>
</string>
```

priority Tags in the Previous and New Templates

In the previous template, `priority` was defined as Low, Normal, or High. The new template sends `priorityCode` instead, which represents the same concept but is sent using an integer representation.

Previous Template	New Template
priority=Low	priorityCode=0
priority=Normal	priorityCode=1
priority=High	priorityCode=2

For example, if you had the following `Normal` mapping inside the `priority` tags of your previous template:

```
<priority>
  <xsl:choose>
    ...
    <xsl:when test="a:SystemAttributes/a:SeverityCode = 'WARNING'">Normal</xsl:when>
    ...
  </xsl:choose>
</priority>
```

The new template should contain the following mapping:

```
<string name="priorityCode">
  <xsl:choose>
    ...
    <xsl:when test="emcf:SystemAttributes/emcf:SeverityCode = 'WARNING'">1</xsl:when>
    ...
  </xsl:choose>
</string>
```

Note that the above translation is assuming that you are customizing the new template to use the old connector's mapping relationship between `priority` and `severity`. The new template's default behavior is to map SCOM's `priority` from the Oracle Enterprise Manager source's `priority`, rather than its `severity`.

CustomField Variable in the Previous and New Templates

In the previous template, if you were sending a `CustomField` variable, it should now be sent as an annotation to the `Message` variable. For example, if you had a `CustomField` variable you were sending in the previous template:

```
<extended-fields>
  <string-field name="CustomField1">MY_CUSTOM_VALUE</string-field>
</extended-fields>
```

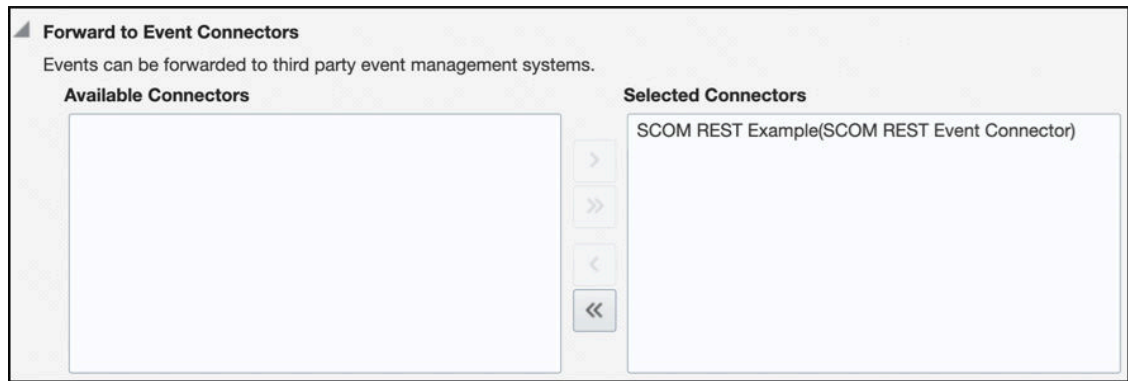
In the new template, it should be annotated with the `Message` variable:

```
<string name="message">
  <xsl:value-of select="emcf:SystemAttributes/emcf:Message" />.
  MY_CUSTOM_VALUE
</string>
```

Forwarding Events to the New Connector

When upgrading from the older connector, you must ensure that you have modified your **Incident Rules** to forward event updates to the new connector instance.

To do so, replace your old connector instance with the new one in this section of the **Add Actions** page of any relevant incident rules:



For more information, see *Using Incident Management in Oracle Enterprise Manager Monitoring Guide*.

Uninstalling the Alert Creator Management Pack

1. In the **Administration** pane of the Microsoft SCOM Operations console, select **Administration**, then **Management Packs**.
2. In the **Management Packs** pane, right click the **Oracle Enterprise Manager Alert Creator** management pack (`OracleEnterpriseManager.Alert.Creator.xml`) and click **Delete**.
3. On the message stating that deleting the management pack might affect the scoping of some user roles, click **Yes**.

Uninstalling the Oracle SCOM Agent

1. Open the Control Panel and go to **Programs and Features** (or **Add or Remove Programs**, depending on your version).
2. Find and select **SCOM Agent**. Select **Uninstall**, and click **Yes**, to confirm.
3. Manually navigate to the installation directory and remove any remaining installation files. The default installation directory for the agent is:

```
C:\iWaveSoftware\
```
4. (Optional) Remove or disable the domain Agent account created exclusively for the Oracle SCOM Agent.

Uninstalling the Microsoft SCOM Web Service on UNIX

1. Run the `service.sh status` command to determine whether the web service is running.
2. If the web service is running, run the `service.sh stop` command to stop the web service and verify it completes successfully.
3. Delete all files in the installation directory.

Uninstalling the Microsoft SCOM Web Service on Windows

1. If the web service is installed as a Windows service:
 - a. Determine if the web service is running.

- b. If the web service is running, run the `service.bat stop` command to stop the web service and verify it completes successfully.
 - c. Run the `service.bat uninstall` command to remove it as a Windows service and verify it completes successfully.
 2. If the web service is *not* installed as a Windows service, perform the following steps:
 - a. Determine if the web service is running.
 - b. If the web service is running, stop the web service by closing the Java window.
 3. Delete all files in the installation directory.

Uninstalling the Microsoft SCOM Event Connector

To uninstall the connector, you must first delete all defined instances of the connector, then you must delete the connector from the **Self Update** page in Oracle Enterprise Manager.

1. From the **Setup** menu, select **Extensibility**, then **Management Connectors**.
2. Select an instance of the connector you want to delete, then click **Delete**.
3. On the **Confirmation** page, click **Yes**.
4. Repeat steps 2 and 3 until all instances of the connector have been deleted.
5. From the **Setup** menu, select **Extensibility**, then **Self Update**.
6. Click the **Management Connector** link in the **Type** column. A list of updates is displayed for Management Connectors.
7. Click the connector you want to delete, select **Actions**, then select **Delete**. The **Delete Update** dialog box is displayed.
8. Click **Delete** to delete the connector. A pop-up confirmation message is displayed.
9. Click **OK** to confirm and delete the connector.