Oracle® Enterprise Manager

Connectors Integration Guide

12*c* Release 2 (12.1.0.2)

E25163-02

October 2012

Oracle Enterprise Manager Connectors Integration Guide, 12*c* Release 2 (12.1.0.2)

E25163-02

Copyright © 2012 Oracle and/or its affiliates. All rights reserved.

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 or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

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

Intel and Intel Xeon 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, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software and documentation may provide access to or information on 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. 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.

Contents

Preface vii

Audience vii
Documentation Accessibility vii
Related Documentation vii
Conventions viii

1 Building a Help Desk Connector

Introduction 1-1
Building a Ticketing Connector 1-2
Defining Connector Descriptor File 1-2
Service Definition 1-6
Template Definition 1-7
Template Registration 1-7
Packaging and Deploying the Connector 1-9

2 Building an Event Connector

Introduction 2-1
Building an Event Connector 2-2
Defining the Connector Descriptor File 2-2
Service Definition 2-4
Template Definition 2-5
Template Registration 2-5
Packaging and Deploying the Connector 2-7

3 Building a Data Exchange Connector

Introduction 3-1
Enterprise Manager and External Management System 3-2
Data Forwarding Frequency Options and Modes 3-2
Data Exchange Concepts 3-3
Data Exchange Hub 3-3
Inbound Data Exchange Session 3-3
Outbound Data Exchange Session 3-3
Normalized Message Format 3-4
Denormalized Message Format 3-4
Message Schemas 3-4
Data Source 3-4
Average Data 3-5
Setting up a Data Exchange Connector 3-5
Creating a Data Exchange Hub 3-5
Using a Third-Party JMS Server as a Data Exchange Hub 3-6
Creating an Outbound Data Exchange Session 3-7

Outbound JMS Destinations 3-8
Outbound Message Schema 3-13
Normalized Message Format 3-13
Denormalized Message Format 3-22
Tuning Outbound Session Message Parameters 3-27
Creating an Inbound Data Exchange Session 3-28
Inbound JMS Destinations 3-31
Inbound Message Schemas 3-31
Inbound Indicators Schema 3-31
Message Semantics 3-31
Inbound Alert Schema 3-33
Integrating Enterprise Manager with OBAM 3-33
Supported Versions 3-33
Setting up the Data Flow from Enterprise Manager to OBAM 3-34
Importing OBAM Artifacts for an Outbound Session 3-34
Updating JNDI 3-38
Setting up the Data Flow from OBAM to Enterprise Manager 3-39
End-to-End Flow 3-39
Using an OC4J as a Data Exchange Hub 3-40
Tips and Troubleshooting Information 3-42
Data Exchange Hub Connection Errors 3-42
Notification Methods and Rules 3-43
Data Flow Tips 3-43
Log Files 3-44
End-to-End Flow Sample Demonstrations 3-45
Suggested Reading 3-45

4 Reference Tables

Request Attributes 4-1
Response File Properties for the Windows Platform 4-9
sl_OHPartitionsAndSpace_valueFromDlg Property 4-9
ret_PrivIntrList Property 4-10
Queryable Properties 4-11
Complex Response Properties 4-14
Status Codes 4-15

A Ticketing Connector Samples

B Error Messages and Debugging

Error Messages B-1
Debugging B-4
Specifying the Debug Option B-4
Viewing Debug Messages B-4

C MOM Event Connector Samples

Glossary

Index

Preface

This Preface contains these sections:

- Audience
- Documentation Accessibility
- Related Documentation
- Conventions

Audience

The Oracle Enterprise Manager Integration Guide is intended for system integrators who want to integrate other management systems with Enterprise Manager.

> **Note:** For the most current version of this document, go to the **Extensibility** page of the Oracle Enterprise Manager Online Documentation set:
>
> `http://www.oracle.com/pls/em121/homepage`

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 Documentation

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

`http://www.oracle.com/technetwork/documentation/`

Oracle Enterprise Manager also provides extensive online help. Click **Help** on any Oracle Enterprise Manager page to display the online Help system.

Printed documentation is available for sale in the Oracle Store at

`http://oraclestore.oracle.com/`

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://www.oracle.com/technetwork/community/join/`

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://www.oracle.com/technetwork/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.
italic	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 Building a Help Desk Connector

This chapter provides information you need to build a help desk connector and integrate it with Enterprise Manager. This chapter has following sections:

- Introduction
- Building a Ticketing Connector
- Defining Connector Descriptor File
- Service Definition
- Template Definition
- Template Registration
- Packaging and Deploying the Connector

1.1 Introduction

Enterprise Manager Cloud Control 12*c* provides a Management Connector Framework (referred to as Connector Framework) to allow developers to build help desk connectors based on metadata (XMLs and XSLs).

Help desk connectors created through the framework inherit the following features:

- **Auto Ticketing —** Lets you describe the connector to automatically open or update a ticket whenever an Incident is triggered in Enterprise Manager. You can specify incident rules for tickets to be created or updated.
- **Manual Ticketing —** Lets you manually create a ticket from Enterprise Manager console based on an open incident in Enterprise Manager. The connector populates the ticket with details based on the incident and the ticket template.

To utilize these ticketing features for your own help desk system, you need to provide a set of metadata files. The categories of metadata files listed in Table 1–1 are required for building a help desk connector:

Table 1–1 Metadata File Categories

Category	Type	Description
Connector Descriptor	XML	Lets you customize the integration for your ticketing system. Through XML you describe how the configuration UI pages should be generated in this file.

Table 1–1 (Cont.) Metadata File Categories

Category	Type	Description
Ticket Request Templates	XML or XLS	Describes how a ticket is filled out in the context of an event. Ticket templates explain the mappings from Enterprise Manager incident data fields to the corresponding ticket data fields in ticket creation and update through the XSL/XML language.
Response Templates	XSL	Normalize the ticketing system Web service's response data into a format that the Ticketing Connector Framework can understand.

1.2 Building a Ticketing Connector

Building a ticketing connector is a two step process that involves writing a deployment descriptor and preparing the appropriate templates to manage the data. The deployment descriptor contains information on the following:

- Basic configuration parameters required by enterprise manager.
- Specification for web services that define how the connector framework communicates with external systems
- Specification for Service Methods

  Service Methods are used to bind Method name with web service endpoints. CreateTicket, updateTicket and getTicket are the supported method names for ticketing connector.
- Specification for Template Registration

  A template is registered based on the information provided in this element. A connector deployment descriptor can have an optional list of up to 50 template registration elements.
- Specification for SOAP header authentication

> **Note:** The name of the connector descriptor must be `connectorDeploy.xml`

1.3 Defining Connector Descriptor File

Define a connector descriptor XML file to describe the connector metadata and the configuration properties of the connector, such as Web service end points and authentication schema.

The key points to remember when constructing a descriptor are

- The connector descriptor file name must be `connectorDeploy.xml`
- The XML file should adhere to schema connectorDeploy.xsd

Refer to the sample `connectorDeploy.xml` in Appendix A, "Ticketing Connector Samples" for reference implementation.

The following table provides a breakdown of the sections of conenctorDeploy.xsd and explains what each section does:

Table 1–2 Connector Descriptor Metadata Details

Metadata Section	Explanation
<SOAPHeaderAuthentication> <Username required="true"> <VariableName>USERNAME</VariableName> <DisplayName>Remedy Username</DisplayName> </Username> <Password> <VariableName>PASSWORD</VariableName> <DisplayName>Remedy Password</DisplayName> </Password> <AuthVariable> <VariableName>AUTHENTICATION</VariableName> <DisplayName>Authentication</DisplayName> </AuthVariable> <AuthVariable> <VariableName>LOCALE</VariableName> <DisplayName>Locale</DisplayName> </AuthVariable> <AuthVariable> <VariableName>TIMEZONE</VariableName> <DisplayName>Timezone</DisplayName> </AuthVariable> <SOAPHeader> <![CDATA[<urn:AuthenticationInfo xmlns:urn="urn:HelpDesk_Submit_Service"> <urn:userName>$USERNAME$</urn:userName> <urn:password>$PASSWORD$</urn:password> <urn:authentication>$AUTHENTICATION$</urn:authentication> <urn:locale>$LOCALE$</urn:locale> <urn:timeZone>$TIMEZONE$</urn:timeZone> </urn:AuthenticationInfo>]]> </SOAPHeader> </SOAPHeaderAuthentication>	(Optional) This section allows you to describe the authentication schema of external system Web services. The authentication section of the Configure Management Connector page in EM is generated based on this section. The username, password, and additional text fields are populated based on this section. - The <VariableName> tag marks the name of the attribute in the Web services XML document. - The <DisplayName> tag describes how the corresponding text-box should be labeled.
<Name>Remedy Service Desk Connector</Name> <Version>12.1.0.1.0</Version> <EMCompatibleVersion>12.1.0.1.0</EMCompatibleVersion> <Description>Remedy Integration with Enterprise Manager</Description> <Category>TicketingConnector</Category>	The connector type name & version element value combination uniquely define a connector instance in EM, hence proper consideration needs to be given while defining the same. The element EMCompatibleVersion is equals to <release specific value> Example, 12.1.0.1.0 for 12C release. oThe element category for external system integration can take one of the following values: EventConnector, TicketingConnector In this case, the value will be TicketingConnector.

Table 1–2 (Cont.) Connector Descriptor Metadata Details

Metadata Section	Explanation

`<ConnectivityTestVariable>`
`<VariableName>TICKET_ID</VariableName>`
`<DisplayName>Ticket ID</DisplayName>`
`</ConnectivityTestVariable>`
	The ConnectivityTestVaiable is a required element for ticketing
	The variable name defined over is used by getTicket method during initial configuration step to validate the connectivity.

`<ExternalURL>`
`<Pattern>`
`<![CDATA[http://$WEB_SERVER$/arsys/forms/$ARSERVER_NAME$/$FORM_NAME$/?qual=%27Incident%20Number*%27=%22@Incident_Number@%22]]>`
`</Pattern>`
`<ConfigVariable required="true">`
`<VariableName>WEB_SERVER</VariableName>`
`<DisplayName>Web Server</DisplayName>`
`</ConfigVariable>`
`<ConfigVariable required="true">`
`<VariableName>FORM_NAME</VariableName>`
`<DisplayName>HelpDesk Case Form Name</DisplayName>`
`</ConfigVariable>`
`<ConfigVariable required="true">`
`<VariableName>ARSERVER_NAME</VariableName>`
`<DisplayName>ARServer Name</DisplayName>`
`</ConfigVariable>`
`</ExternalURL>`
	This section enables you to configure the Ticket Connector Framework to generate a ticket URL to your unique Ticketing System.
	The value of the <Pattern> tag describes the URL string, and how user configured variables like ARSERVER_NAME,FORM_NAME and WEB_SERVER are inserted into it.
	A textbox label pair is inserted into the Webconsole section for each <ConfigVariable> tag. (Figure 1-2)
	The values that the users provide for each user variable is inserted into the URL pattern string accordingly. If there is a user variable "X" then the user input value replaces "[X]" when the ticket URL is generated.

`<Service>`
`<Method>createTicket</Method>`
`<WebServiceEndpoint>`
`<![CDATA[http://[midtier_server]/arsys/services/ARService?server=[servername]&webService=HPD_IncidentInterface_Create_WS]]>`
`</WebServiceEndpoint>`
`</Service>`
`<Service>`
`<Method>updateTicket</Method>`
`<WebServiceEndpoint>`
`<![CDATA[http://[midtier_server]/arsys/services/ARService?server=[servername]&webService=HPD_IncidentInterface_WS]]>`
`</WebServiceEndpoint>`
`</Service>`
`<Service>`
`<Method>getTicket</Method>`
`<WebServiceEndpoint>`
`<![CDATA[http://[midtier_server]/arsys/services/ARService?server=[servername]&webService=HPD_IncidentInterface_get_WS]]>`
`</WebServiceEndpoint>`
`</Service>`
	This section allows you to specify configurations specific to the Ticketing System's Web services.
	The <WebServiceEndpoint> tag describes the default Web service endpoint string to be displayed in the Web service section of the Management Connector page.
	Method defines one of the EM specific service name
	SoapAction defines the operation that needs to be invoked at external system
	Refer to the Service Definition Section for more details.

Table 1–2 (Cont.) Connector Descriptor Metadata Details

Metadata Section	Explanation

`<TemplateRegistration>`
`<FileName>getTicket_request.xml</FileName>`
`<InternalName>getTicket</InternalName>`
`<TemplateName>Get Ticket</TemplateName>`
`<TemplateType>OutboundXML</TemplateType>`
`<Description>This is the request xml file for getTicket method</Description>`
`</TemplateRegistration>`
`<TemplateRegistration>`
`<FileName>getTicket_response.xsl</FileName>`
`<InternalName>getTicket</InternalName>`
`<TemplateName>Get Ticket</TemplateName>`
`<TemplateType>InboundXSL</TemplateType>`
`<Description>This is the response xsl file for getTicket method</Description>`
`</TemplateRegistration>`
`<TemplateRegistration>`
`<FileName>createTicket_response.xsl</FileName>`
`<InternalName>createTicket</InternalName>`
`<TemplateName>Create Ticket Reponse</TemplateName>`
`<TemplateType>InboundXSL</TemplateType>`
`<Description>This is the create ticket response template. </Description>`
`</TemplateRegistration>`
`<TemplateRegistration>`
`<FileName>templates/Remedy_DefaultCategory.xsl</FileName>`
`<InternalName>Remedy_DefaultCategory.xsl</InternalName>`
`<TemplateName>Remedy Default Category </TemplateName>`
`<TemplateType>OutboundXSL</TemplateType>`
`<Description>This is the Remdy default category template. </Description>`
`</TemplateRegistration>`
	This section is to add connector templates. There are 2 types of template. 1. Request and 2. response templates for createTicket and getTicket methods.
	Refer to Template Definition section for more details.

Table 1–2 (Cont.) Connector Descriptor Metadata Details

Metadata Section	Explanation

<SOAPHeaderAuthentication>
    <Username required="true">
      <VariableName>USERNAME</VariableName>
      <DisplayName>Username</DisplayName>
    </Username>
    <Password>
      <VariableName>PASSWORD</VariableName>
      <DisplayName> Password</DisplayName>
    </Password>
    <AuthVariable>
<VariableName>AUTHENTICATION</VariableName>
<DisplayName>Authentication</DisplayName>
    </AuthVariable>
    <AuthVariable>
      <VariableName>LOCALE</VariableName>
      <DisplayName>Locale</DisplayName>
    </AuthVariable>
    <AuthVariable>
     <VariableName>TIMEZONE</VariableName>
      <DisplayName>Timezone</DisplayName>
    </AuthVariable>
    <SOAPHeader>
     <![CDATA[
     ]]>
    </SOAPHeader>
  </SOAPHeaderAuthentication>
	(Optional) This section allows you to describe the authentication schema of external system Web services.
	The authentication section of the Configure Management Connector page in EM is generated based on this section.
	The username, password, and additional text fields are populated based on this section.
	The <*VariableName*> tag marks the name of the attribute in the Web services XML document.
	The <*DisplayName*> tag describes how the corresponding text-box should be labeled.

1.4 Service Definition

Following event connector service operations are supported:

- createTicket

  Create ticket on the external system.

- updateTicket

  Forward event updates to the external system.

- getTicket

  Service method to validate the ticket connector connectivity with Enterprise Manager.

Note that the service names in the connector descriptor should exactly match the names defined above and are case sensitive.

The following is an example of service setup:

```
<Service>
<Method>createTicket</Method>
<WebServiceEndpoint>
<![CDATA[http://[midtier_
server]/arsys/services/ARService?server=[servername]&webService=HPD_
IncidentInterface_Create_WS]]></WebServiceEndpoint>
</Service>
```

1.5 Template Definition

Templates contain data that gets transformed while sending request and receiving response. A template can be defined as XML or XSL file for a particular service operation in the Enterprise Manager.

A template implementation for ticketing connector can be further classified as:

Table 1–3 Ticket Implementation

Implementation Type	Description	Example

Service method	These templates are applicable to defined service method only.	`<TemplateRegistration>`
		`<FileName>getTicket_request.xml</FileName>`
		`<InternalName>getTicket</InternalName>`
		`<TemplateName>Get Ticket</TemplateName>`
		`<TemplateType>OutboundXML</TemplateType>`
		`<Description>This is the request xml file for getTicket method</Description>`
		`</TemplateRegistration>`
		`<TemplateRegistration>`

Default	These templates can be used by either of createTicket or updateTicket request service methods.	`<TemplateRegistration>`
		`<FileName>templates/Remedy_DefaultCategory.xsl</FileName>`
		`<InternalName>Remedy_DefaultCategory.xsl</InternalName>`
		`<TemplateName>Remedy Default Category </TemplateName>`
		`<TemplateType>OutboundXSL</TemplateType>`
		`<Description>This is the Remdy default category template. </Description>`
		`</TemplateRegistration>`

The following are the guidelines for defining any template:

- Method specific templates should have internal name same as the method name (case sensitive)
- Default templates can have any internal names.
- The template type should be:

  ```
  InboundXSL for response xsl,
  OutboundXSL for request xsl, and
  OutboundXML for request xml
  ```

- The template filename is no longer important, but Oracle recommends that you use the following:

  ```
  <methodName>_request.xml,
  <methodName>_request.xsl, and
  <methodName>_response.xsl.
  ```

1.6 Template Registration

Coding template registration XML or XSL is driven by the following:

- Defined set of schema definition (XSD)
- Target external system attributes

The following table lists the schema definitions for Enterprise Manager Event Connector:

Table 1–4 Schema Definitions for Enterprise Manager Event Connector

Name	**Description**
EMIncident.xsd	It defines an EM incident made available through the connector framework
ConnectorCommon.xsd	It defines all the data types used in all the other xsds.For example, SourceObjInfoType
createTicket_response.xsd	It provides the schema of the response for createTicket and updateTicket response
getTicket_response.xsd	It provides the schema of the response for a getTicket method
PublishTicket.xsd	It provides the schema for publishTicket request operation

Key points for defining the request templates are

- The getTicket service only requires the request XML template file to be defined.

  Template request XML is used if the information to be passed is already defined as connector test variable element. The connector test variable can be defined via Connector Deployment Descriptor XML at the time of registration.

- The createTicket and updateTicket services use default XSL template files defined via connector deployment descriptor XML.

  Template request XSL is used to generate a request with event and incident attributes generated within Enterprise Manager. These request require mapping within Enterprise Manager event and incident objects and ticketing system attributes.

- Defining the template Request XSL is optional if the user has already defined Request XML.

  Ideally there won't be any need for this scenario.

- You can use the EMIncident.xsd and EMEvent.xsd schema to decide which Enterprise Manager fields you want to make available for mapping when creating a ticketing request.

Key points for defining the response templates are

- You can define the response template(XSL) based on the XSD defined in the Ticketing Connector Template XSD table.

- Template response XSL are optional.

  If no response transformation(XSL) is defined, then any data returned by external systems within response XML will not be persisted. This is the ideal implementation for operations that does not care about external system response data.

- For createTicket and updateTicket services, you can use the createTicket_response.xsd schema to decide how to transform the response from external systems to the response understandable by Enterprise Manager.

After the transformation, Enterprise Manager expects a <TicketId> and updates the proper status depending on the value:

- Not Null

  Enterprise manager will persist the ticketId with the associated Incident and annotate the proper status messages in the EM Incident Manager grid.

- Null

  Enterprise manager will annotate a error message with the associated Incident in the EM Incident Manager grid.

Refer to following sample template files for Remedy 7 Ticketing Connector implementation:

Table 1–5 Sample Template Files

Service Name	Request XSL	Request XML	Response XSL
getTicket	optional	getTicket_request.xml	getTicket_response.xsl
CreateTicket /updateTicket	Remedy_ DefaultCategory.xsl Remedy_ DefaultAutoClose.xsl Remedy_ DefaultAutoResolve.xsl	Not applicable	createTicket_response.xsl

1.7 Packaging and Deploying the Connector

To deploy the connector, Enterprise Manager uses the Self Update feature. This feature, which can be accessed through the console, provides the ability to import the connector into the Enterprise Manager environment. To deploy the connector complete the following:

1. Prepare the connector jar file

   Package all XML and XSL template files as a .jar file

   ```
   <name>_connector.jar
   ---> connectorDeploy.xml
   --->template1.xml
   --->template2.xsl
   …..
   …..
   --->templateN.xsl
   ```

2. Prepare the manifest file

   Key attributes of the self update manifest files are

   - EntityType

     Value is core_connector

   - EntityTypeVersion

     Current release version. Value=12.1.0.1.0

   - Attribute @Name=connector_type

     Connector type name

- Attribute @Name=connector_category

  Category type can be TicketingConnector or EventConnector

- ArchiveList

  This element contains the list of archives that are part of the connector setup. Generally there will be a single connector jar, but for some special implementations there may be additional jars (adapter or agent). In these cases, the connector specific jar should be the first one in the defined list. This is a mandatory requirement.

The following example shows the code for the `connector_manifest.xml` file.

```
<EntityInstanceList
xmlns="http://www.oracle.com/EnterpriseGridControl/SelfUpdateManifest">
 <EntityInstance
xmlns="http://www.oracle.com/EnterpriseGridControl/SelfUpdateManifest"
EntityTypeVersion="11.2.0.1.0" EntityType="core_connector"
Maturity="PRODUCTION" Vendor="Oracle" PluginID="oracle.sysman.core">
 <Description>
 <![CDATA[ BMC Remedy Service Desk Connector - 12.1.0.2.0
  ]]>
  </Description>
 <AttributeList>
  <Version>12.1.0.2.0</Version>
  <Attribute Name="connector_type" Value="Sample SCOM Connector" Label="Sample
SCOM Connector" />
  <Attribute Name="connector_category" Value="EventConnector" Label="Event
Connector" />
  </AttributeList>
 <Readme>
 <![CDATA[   Oracle Management Connector for SCOM integrates Oracle Enterprise
Manager Cloud Control's proactive alert detect
ion and resolution features with BMC's Remedy 7.0 Service Desk capabilities to
provide a seamless workflow for incident mana
gement and resolution.
  Change Logs:
  12.1.0.2.0
   - Miscellaneous bug fixes
   - Performance enhancements
  ]]>
  </Readme>
  <DependsOn />
 <ArchiveList>
  <Archive Filename="scomconnector.jar" IsMDS="false" />
  <Archive Filename="SCOM_webservices_adapter.jar" IsMDS="false" />
  </ArchiveList>
 <CustomData>
 <![CDATA[
  ]]>
  </CustomData>
 </EntityInstance>
 </EntityInstanceList>
```

Refer to the self update schema definition from following location for the complete list of attributes:

```
$ORACLE_HOME/sysman/emSDK/core/selfupdate/model/SelfUpdateManifest.xsd
```

3. Configure emedk tool

   The emedk tool can be configured by following instructions from EM UI. The path is Setup > Extensibility > Development Kit.

4. Prepare the self-update archive

   This requires the connector jar file and the manifest file for the connector. To prepare self-update, call the following utility to create a self update archive file:

   ```
   edktool prepare_update
           -manifest "manifest xml"
           -archivedir "archives directory"
           -out "output file or directory"
           [-typexml "update type xml"]
   ```

   Table 1–6 describes the options available with the utility

Table 1–6 Self Update Utility Options

Option	Description
`-manifest`	Self update manifest file that describes the update.
`-archivedir`	Directory containing the archive files specified in the manifest file.
`-out`	Directory or filename for the self update archive. If a directory is specified, the filename is autogenerated.
`-typexml`	Optional path to the update type xml

   The following example creates a self update archive in the `/u01/sar` directory based on the manifest file `/u01/connector/connector_manifest.xml`. The archives referred to in `connector_manifest.xml` are picked from the directory `/u01/connector/archives`.

   ```
     edkUtil prepare_update
             -manifest /u01/connector/connector1_manifest.xml
             -archivedir /u01/connector/archives
             -out  /u01/sar/sample_connector.zip
   ```

5. Import the connector archive to Enterprise Manager by calling any one of the following `emcli` commands:

   ```
   edktool import_update
   -file=\ file\
   -omslocal

   emcli import_update
   -file=\ file\
   -host=\ hostname\
   [-credential_set_name=\ setname\ ] | -credential_name=\ name\  -credential_
   owner=\ owner\
   ```

   These commands import a Self Update archive file into Enterprise Manager. On successful import, the update is displayed on the Self Update Home in downloaded status for further action.

Table 1–7 describes the options available with this command:

Table 1–7 Connector Archive Command Options

Options	Description
-file	The complete pathname of the update archive file
-omslocal	The flag specifying that the file is accessible from the OMS
-host	The target name for a host target where the file is available
-credential_set_name	The set name of the preferred credential stored in the repository for the host target. Can be one of the following: ■ `HostCredsNormal` Default unprivileged credential set ■ `HostCredsPriv` Privileged credential set
-credential_name	The name of a named credential stored in the repository. This option must be specified along with -credential_owner option.
-credential_owner	The owner of a named credential stored in the repository. This option must be specified along with -credential_name option.

The following paragraphs provide some examples of the use of the `emcli` command:

Example 1

Imports the file `sample_connector.zip`. The file must be present on the OMS host. In a multiple OMS setup, the request can be processed by any OMS, so the file should be accessible from the OMS processing the request. This usually means that the file must be kept on a shared location that is accessible from all OMS.

```
emcli import_update
  -file=\ /u01/sar/sample_connector.zip
  -omslocal
```

Example 2

Imports the file `sample_connector.zip.zip` that is present on the `host1.example.com` host. The host must be a managed host target in Enterprise Manager and the agent on this host must be up and running. The preferred unprivileged credentials for host `host1.example.com` are used to retrieve the remote file.

```
emcli import_update
  -file=\ /u01/sar/sample_connector.zip
  -host=\ host1.example.com\
  -credential_set_name=\ HostCredsNormal\
```

Example 3

Imports the file `sample_connector.zip` that is present on the `host1.example.com` host. The host must be a managed host target in Enterprise Manager and the agent on this host must be up and running. The named credentials `\ host1_creds\` owned by user `\ admin1\` are used to retrieve the remote file.

```
emcli import_update
  -file=\ /u01/sar/sample_connector.zip\
  -host=\ host1.example.com\
  -credential_name=\ host1_creds\
  -credential_owner=\ admin1\
```

6. Go to Self-Update Home page

   The connector will be shown as downloaded.

7. Select the connector row and click **Apply** to deploy the connector.

2 Building an Event Connector

This chapter provides information that you need to build an event connector and integrate it with Enterprise Manager.

This chapter has the following sections:

- Introduction
- Building an Event Connector
- Defining the Connector Descriptor File
- Service Definition
- Template Definition
- Template Registration
- Packaging and Deploying the Connector

2.1 Introduction

Enterprise Manager Cloud Control 12*c* provides a Management Connector Framework (referred to as Connector Framework) that enables developers to build event connectors based on metadata (XMLs and XSLs). The event connector allows Enterprise Manager to send generated events to external systems.

The following metadata categories are required to build an event connector:

- Connector Descriptor XML

  Provides the definition of connector in XML form
- Request Templates XML or XSL

  Provide transformation mapping for EM event and external system attributes using predefined XML template or XSL.
- Response Template XSL

  Transform response from external system into EM data model.

Note: The current release only supports outbound operations (sending events to external systems). The support for inbound (importing) external events into Enterprise Manager may be considered for future release.

2.2 Building an Event Connector

Building an event connector is a two step process that involves writing a deployment descriptor and preparing the appropriate templates to manage the data. The connector descriptor contains information on the following:

- Basic configuration parameters required by enterprise manager.
- Specification for Service methods

  Service methods are used to bind the method name with Web service endpoints that define how the connector framework communicates with external systems.
- Specification for template registration

  A template is registered based on the information provided in the element. A connector deployment descriptor can have an optional list of up to 50 template registration elements.
- Specification for SOAP header authentication (optional)

> **Note:** In an upgrade scenario, all external alerts migrated to 12c as Enterprise Manager events will not be shown on the target page. Customers need to manually create Incidents for these events so that they can be shown on the target page.
>
> In a fresh 12c install scenario, inbound operations are not supported so no external events will be generated and nothing will be shown on the target page.

2.3 Defining the Connector Descriptor File

Define a connector descriptor XML file to describe the connector metadata and the configuration properties of the connector, such as Web service end points and authentication schema.

Key points to remember while constructing the constructor descriptor:

- The connector descriptor file name must be `connectorDeploy.xml`
- The XML file should adhere to schema `connectorDeploy.xsd`

Refer to the sample `connectorDeploy.xml` in Appendix C, "MOM Event Connector Samples" for reference implementation. The following table provides a break-down of the sections of `conenctorDeploy.xsd` and explains what each section does.

Table 2–1 Connector Descriptor Metadata Details

Metadata Section	Description

<Name>Microsoft Operations Manager Connector</Name>
  <Version>12.1.0.1.0</Version>
<EMCompatibleVersion>12.1.0.1.0</EMCompatibleVersion>
  <Description>Microsoft Operations Manager Integration with Enterprise Manager</Description>
  <Category>EventConnector</Category>
  <ConfigVariable required="true">
    <VariableName>SETUP_NAME</VariableName>
    <DisplayName>MOM Registered Connector Name</DisplayName>
  </ConfigVariable>
  <ConfigVariable required="true">
    <VariableName>RESOLUTION_STATE</VariableName>
    <DisplayName>Resolution State</DisplayName>
  </ConfigVariable>
	The connector type name & version element value combination uniquely define a connector instance in EM, hence proper consideration needs to be given while defining the same.
	The element EMCompatibleVersion is equals to <release specific value>.
	Example, 12.1.0.1.0 for 12C release.
	The element category for external system integration can take one of the following values:
	EventConnector
	TicketingConnector
	In this case, the value will be EventConnector.
	The ConnectivityTestVariable & ExternalURL elements are not applicable for event connector.

<Service>
    <Method>cleanup</Method>
    <WebServiceEndpoint>
<![CDATA[http://[server]:1271/ConnectorServiceV2.asmx]]>
    </WebServiceEndpoint>
<SOAPAction>http://www.microsoft.com/EnterpriseManagement/Mom/Connector/V2/Cleanup</SOAPAction>
  </Service>
	Method defines one of the EM specific service name.
	Web service end point defines the external system url.
	SoapAction defines the operation that needs to be invoked at external system.
	Refer to the Service Definition Section for more details.

<TemplateRegistration>
    <FileName>setup_request.xml</FileName>
    <InternalName>setup</InternalName>
    <TemplateName>Setup</TemplateName>
<TemplateType>OutboundXML</TemplateType>
    <Description>This is the request xml file for setup method</Description>
  </TemplateRegistration>
	Template Registration defines the definition of XML or XSL template.
	Refer to the Template Definition section for more details.

Table 2–1 (Cont.) Connector Descriptor Metadata Details

Metadata Section	Description

`<SOAPHeaderAuthentication>`
`    <Username required="true">`
`      <VariableName>USERNAME</VariableName>`
`      <DisplayName>Username</DisplayName>`
`    </Username>`
`    <Password>`
`      <VariableName>PASSWORD</VariableName>`
`      <DisplayName> Password</DisplayName>`
`    </Password>`
`    <AuthVariable>`
 	
`<VariableName>AUTHENTICATION</VariableName>`
 	
`<DisplayName>Authentication</DisplayName>`
`    </AuthVariable>`
`    <AuthVariable>`
`      <VariableName>LOCALE</VariableName>`
`      <DisplayName>Locale</DisplayName>`
`    </AuthVariable>`
`    <AuthVariable>`
`      <VariableName>TIMEZONE</VariableName>`
`      <DisplayName>Timezone</DisplayName>`
`    </AuthVariable>`
 	
`    <SOAPHeader>`
`     <![CDATA[`
`     ]]>`
`    </SOAPHeader>`
`  </SOAPHeaderAuthentication>`
	(Optional) This section allows you to describe the authentication schema of external system Web services.
	The authentication section of the Configure Management Connector page in EM is generated based on this section.
	The username, password, and additional text fields are populated based on this section.
	■ The <VariableName> tag marks the name of the attribute in the Web services XML document.
	■ The <DisplayName> tag describes how the corresponding text-box should be labeled.

2.4 Service Definition

The event connector service operations defined in the following table are supported in the current release:

Table 2–2 Connector Service Operations

Name	Description
setup (optional)	Register a connector with external system
Initialize (optional)	Initialize a connector with external system after setup
Uninitialize (optional)	Un-initializes the connector with the external event system
Cleanup (optional)	De-registers the connector with the external event system.
createEvent	Forward a new event to external system
updateEvent	Forward event updates to external system

The following are the key points for the above mentioned services:

- The service names in the connector descriptor should exactly match the names defined in Table 2–2, " Connector Service Operations" (case sensitive).
- Setup and initialize are called in the order when the connector is configured if they're defined.
- Uninitialize and cleanup are opposite actions to initialize and setup.

Uninitialize can be defined if initialize is defined. Cleanup can be defined if setup is defined. If defined, the services are called in the order of uninitialize and cleanup when the connector is deleted.

- createEvent and updateEvent are called to forward an event from Enterprise Manager to an external system generated through event rules.

The following is an example for setup service:

```
<Service>
   <Method>setup</Method>
   <WebServiceEndpoint>....</WebServiceEndpoint>
   <SOAPAction>.......</SOAPAction>
</Service>
```

2.5 Template Definition

Templates contains data that gets transformed while sending a request and receiving a response. A template can be defined as an XML or XSL file for a particular service operation in Enterprise Manager.

The following are the guidelines for defining any template:

- The internal name should be the same as the method name (case sensitive)
- The template type should be:

```
 InboundXSL for response xsl,
 OutboundXSL for request xsl, and
 OutboundXML for request xml
```

- The template filename is no longer important, but we recommend using the following:

```
<methodName>_request.xml,
<methodName>_request.xsl, and
<methodName>_response.xsl.
```

  For example:

```
<TemplateRegistration>
    <FileName>setup_request.xml</FileName>
    <InternalName>setup</InternalName>
    <TemplateName>Setup</TemplateName>
    <TemplateType>OutboundXML</TemplateType>
   <Description>This is the request xml file for setup method</Description>
 </TemplateRegistration>
```

2.6 Template Registration

Coding template registration XML or XSL is driven by the following:

- Defined set of schema definition (XSD).
- Target external system attributes.

Table illustrate the list of schema definition for Enterprise Manager Event Connector.

Table 2–3 Event Connector Template XSD

Name	Description
EMEvent.xsd	Defines an EM event made available through the connector framework.
ConnectorCommon.xsd	Defines all the data types used in all the other xsds, for example, SourceObjInfoType.
EMEventResponse.xsd	Defines the response of the createEvent/updateEvent method. The response is from the external server for an Event Manager event it has received.
initialize_response.xsd	Provides the schema of the response for an initialize method.
setupResponse.xsd	Provides the schema of the response for a setup method.
uninitialize_response.xsd	Provides the schema of the response for an uninitialize method.

The following are the key points for defining the request templates:

- The setup, initialize, uninitialize, and cleanup services require request XML template files to be defined.

  The template request XML is used if the information to be passed is already defined as a connector variable element. The connector variable can be defined using:

  - Connector Deployment Descriptor at the time of registration
  - Response message of any of these services.

- The createEvent and updateEvent services require only XSL files to be defined.

  The template request XSL is used to generate a request with event attributes generated within Enterprise Manager. These requests require mapping within Enterprise Manager event object and target external system attributes.

- Defining template request XSLs is optional if you have defined request XML. Ideally there won't be any need for this scenario.
- You can use the EMEvent.xsd schema to decide which Enterprise Manager fields you want to make available for mapping when sending events to external systems.

The following are the key points for defining the response templates:

- You can define the response template (XSL) based on the XSD defined in Table 2–3, " Event Connector Template XSD".
- Template response XSL are optional.

  If no response transformation (XSL) is defined, then any data returned by an external system within the response XML will not be persisted. This is an ideal implementation for an operation not concerned with external system response data.

- For createEvent and updateEvent services, you can use the EMEventResponse.xsd schema to decide how to transform the response from external systems to a response understandable by Enterprise Manager.

After the transformation, Enterprise Manager expects a <SuccessFlag> and one of the following elements depending on the value:

- True

Enterprise Manager expects an <externalEventID>, which is the ID of the event in the external system. This is used to update the event in the future.

- False

  Enterprise Manager expects an <ErrorMessage>. Enterprise Manager annotates the event with the externalEventID or ErrorMessage.

Refer to following sample template files in Table 2–4, " Sample Template List" for reference MOM event connector implementation:

Table 2–4 Sample Template List

Service Name	Request XSL	Request XML	Response XSL
setup	Optional	setup_request.xml	setup_response.xsl
initialize	Optional	initialize_request.xml	Optional
uninitialize	Optional	uninitialize_request.xml	Optional
cleanup	Optional	cleanup_request.xml	Optional
createEvent	createEvent_request.xsl	Not applicable	createEvent_response.xsl
updateEvent	updateEvent_request.xsl	Not applicable	updateEvent_response.xsl

2.7 Packaging and Deploying the Connector

To deploy the connector, Enterprise Manager uses the Self Update feature. This feature, which can be accessed through the console, provides the ability to import the connector into the Enterprise Manager environment. To deploy the connector complete the following:

1. Prepare the connector jar file

   Package all XML and XSL template files as a .jar file

   ```
   <name>_connector.jar
   ---> connectorDeploy.xml
   --->template1.xml
   --->template2.xsl
   …..
   …..
   --->templateN.xsl
   ```

2. Prepare the manifest file

   The following table lists the Key attributes of self update manifest files:

Table 2–5 Self Update Manifest File Attributes

Name	Description
EntityType	Value is core_connector
EntityTypeVersion	Current release version. Value=12.1.0.1.0
Attribute @Name=connector_type	Connector type name
Attribute @Name=connector_category	Category type can be TicketingConnector or EventConnector

Table 2–5 (Cont.) Self Update Manifest File Attributes

Name	Description
ArchiveList	This element contains the list of archives that are part of connector setup. Generally there will be single connector jar but for some special implementation there may be additional jars(adapter or agent). In these cases, the connector specific jar should be first one in the defined list. This is mandatory requirement.

The following example shows the code for the connector_manifest.xml file:

```
<EntityInstanceList
xmlns="http://www.oracle.com/EnterpriseGridControl/SelfUpdateManifest">
 <EntityInstance
xmlns="http://www.oracle.com/EnterpriseGridControl/SelfUpdateManifest"
EntityTypeVersion="11.2.0.1.0" EntityType="core_connector"
Maturity="PRODUCTION" Vendor="Oracle" PluginID="oracle.sysman.core">
 <Description>
 <![CDATA[ BMC Remedy Service Desk Connector - 12.1.0.2.0
  ]]>
  </Description>
 <AttributeList>
  <Version>12.1.0.2.0</Version>
  <Attribute Name="connector_type" Value="Sample SCOM Connector" Label="Sample
SCOM Connector" />
  <Attribute Name="connector_category" Value="EventConnector" Label="Event
Connector" />
  </AttributeList>
 <Readme>
 <![CDATA[   Oracle Management Connector for SCOM integrates Oracle Enterprise
Manager Cloud Control's proactive alert detect
ion and resolution features with BMC's Remedy 7.0 Service Desk capabilities to
provide a seamless workflow for incident mana
gement and resolution.
  Change Logs:
  12.1.0.2.0
   - Miscellaneous bug fixes
   - Performance enhancements
  ]]>
  </Readme>
  <DependsOn />
 <ArchiveList>
  <Archive Filename="scomconnector.jar" IsMDS="false" />
  <Archive Filename="SCOM_webservices_adapter.jar" IsMDS="false" />
  </ArchiveList>
 <CustomData>
 <![CDATA[
  ]]>
  </CustomData>
 </EntityInstance>
 </EntityInstanceList>
```

Refer to the self update schema definition from following location for a complete list of attributes:

```
$ORACLE_HOME/sysman/emSDK/core/selfupdate/model/SelfUpdateManifest.xsd
```

3. Configure the emedk tool

The emedk tool can be configured by following instructions from the Enterprise Manager user interface. The path is Setup > Extensibility > Development Kit.

4. Prepare the self-update archive

   This requires the connector jar file and the manifest file for the connector. To prepare self-update, call the following utility to create a self update archive file:

```
edkUtil prepare_update
        -manifest "manifest xml"
        -archivedir "archives directory"
        -out "output file or directory"
        [-typexml "update type xml"]
```

Table 2–6 Self Update Utility Options

Option	Description
-manifest	Self update manifest file that describes the update.
-archivedir	Directory containing the archive files specified in the manifest file.
-out	Directory or filename for the self update archive. If a directory is specified, the filename is autogenerated.
-typexml	Optional path to the update type xml

   The following example creates a self update archive in the `/u01/sar` directory based on the manifest file `/u01/connector/connector_manifest.xml`. The archives referred to in connector_manifest.xml are picked from the directory `/u01/connector/archives`.

```
  edkUtil prepare_update
          -manifest /u01/connector/connector_manifest.xml
          -archivedir /u01/connector/archives
          -out  /u01/sar/sample_connector.zip
```

5. Import the connector archive to Enterprise Manager by calling any one of the following emcli commands:

```
emcli import_update
  -file=\ file\
  -omslocal
```

   or

```
emcli import_update
-file=\ file\
-host=\ hostname\
[-credential_set_name=\ setname\ ] | -credential_name=\ name\  -credential_
owner=\ owner\
```

   These commands import a Self Update archive file into Enterprise Manager. On successful import, the update is displayed on the Self Update Home in downloaded status for further action. Table 2-7 describes the connector archive command options.

Table 2–7 Connector Archive Command Options

Options	Description
-file	The complete pathname of the update archive file
-omslocal	The flag specifying that the file is accessible from the OMS

Table 2–7 (Cont.) Connector Archive Command Options

Options	Description
-host	The target name for a host target where the file is available
-credential_set_name	The set name of the preferred credential stored in the repository for the host target. Can be one of the following: ■ `HostCredsNormal` Default unprivileged credential set ■ `HostCredsPriv` Privileged credential set
-credential_name	The name of a named credential stored in the repository. This option must be specified along with -credential_owner option.
-credential_owner	The owner of a named credential stored in the repository. This option must be specified along with -credential_name option.

The following paragraphs provide some examples of the use of the `emcli` command:

Example 1

Imports the file `update1.zip`. The file must be present on the OMS host. In a multiple OMS setup, the request can be processed by any OMS, so the file should be accessible from the OMS processing the request. This usually means that the file must be kept on a shared location that is accessible from all OMS.

```
emcli import_update
  -file=\ /u01/common/update1.zip\
  -omslocal
```

Example 2

Imports the file `update1.zip` that is present on the `host1.example.com` host. The host must be a managed host target in Enterprise Manager and the agent on this host must be up and running. The preferred unprivileged credentials for host `host1.example.com` are used to retrieve the remote file.

```
emcli import_update
  -file=\ /u01/common/update1.zip\
  -host=\ host1.example.com\
  -credential_set_name=\ HostCredsNormal\
```

Example 3

Imports the file `update1.zip` that is present on the `host1.example.com` host. The host must be a managed host target in Enterprise Manager and the agent on this host must be up and running. The named credentials `\ host1_creds\` owned by user `\ admin1\` are used to retrieve the remote file.

```
emcli import_update
  -file=\ /u01/common/update1.zip\
  -host=\ host1.example.com\
  -credential_name=\ host1_creds\
  -credential_owner=\ admin1\
```

6. Go to Self-Update Home page

   The connector will be shown as downloaded.

7. Select the connector row and click **Apply** to deploy the connector.

3 Building a Data Exchange Connector

A Data Exchange Connector is a JMS server-based integration vehicle that helps you to build a bi-directional data exchange setup between Enterprise Manager and other management systems. The Data Exchange Connector architecture is based on open standards such as Java Message Service (JMS) and XML. This helps in facilitating easy extensibility and interoperability.

The data exchange environment necessitates creation of a data exchange hub and data exchange sessions. This chapter explains the key concepts, components, and features involved in the data exchange process.

Also provided are specific steps to integrate Enterprise Manager with Oracle Business Activity Monitoring Server (OBAM).

This chapter discusses these topics:

- Introduction
- Data Exchange Concepts
- Setting up a Data Exchange Connector
- Integrating Enterprise Manager with OBAM
- Using an OC4J as a Data Exchange Hub
- Tips and Troubleshooting Information
- Suggested Reading

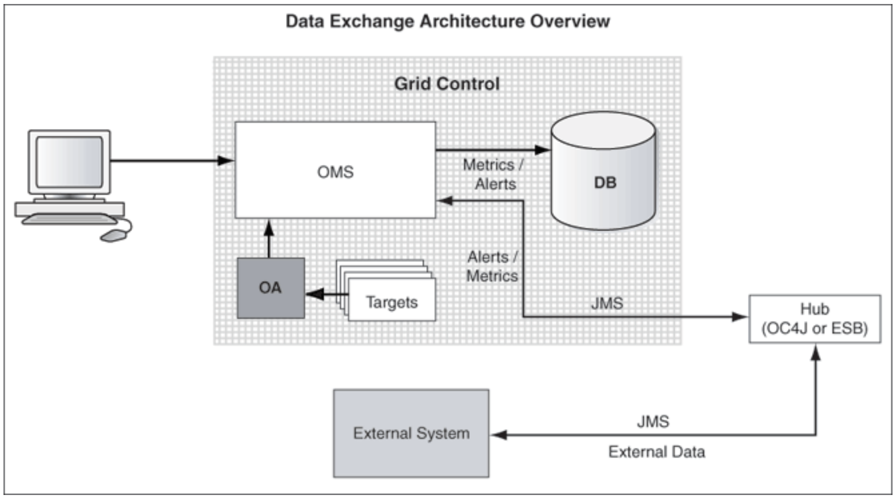
3.1 Introduction

Typically, an enterprise may have Enterprise Manager monitoring most of the systems and services within it. However, other monitoring systems or external management systems such as the OBAM server might also exist within the business environment of the enterprise. These management systems and Enterprise Manager might be collecting monitoring information that is different, yet related to the same business application. It is imperative for the business that Enterprise Manager and these external management systems co-exist and interact seamlessly.

A Data Exchange Connector effectively addresses this communication requirement by transferring data in XML format using JMS Topic or Queue messages. This is made possible by creating a data exchange hub and a data exchange session.

Figure 3–1 provides an architectural overview of the Data Exchange Connector.

Figure 3–1 Data Exchange Architecture



3.1.1 Enterprise Manager and External Management System

Table 3–1 explains the data requirements and purpose of data exchange between Enterprise Manager and an external management system.

Table 3–1 Data Exchange between Enterprise Manager and External Management System

Data Exchange	Requirement	Purpose
From Enterprise Manager to external management system	■ Metric Data ■ Alert Data	■ To use the data to correlate with business indicators and events that business applications send to an external management system. Metric data also includes Service-level Agreement (SLA) and target status data besides raw metrics. ■ Better reporting, event notification, and corrective actions.
From external management system to Enterprise Manager	■ Business Indicators ■ Business Events	■ Better reporting and topology analysis. ■ Generate a comprehensive SLA in the Enterprise Manager environment.

Integrating the two systems using Data Exchange Connector helps the systems to complement each other and serve business requirements effectively and economically.

3.1.2 Data Forwarding Frequency Options and Modes

The following list explains the normal process followed when sending data from Enterprise Manager to external management systems.

- Real-time metric, availability, and SLA values are forwarded, as well as historical data.
- Historical data is forwarded for the last 24 hours, 7 days, and 31 days. Historical data for the target status is sent as the percentage of time in which the target is available during the time period.

- Metric data is forwarded in batches at scheduled intervals. In each batch, a maximum of 100 data points can be sent. An interval of two seconds is maintained between subsequent forwarding to reduce the JMS server load.
- For a given metric, all new data points in the interval are sent to the external system. If there are no new values, no data is sent. For the initial forwarding, data points since the previous one hour are considered. One hour is the default interval, but you can configure this to a different time interval.

  For example, if an outbound session is scheduled from 9:00 a.m. to 9:00 p.m. with an interval frequency of 30 minutes, initially (at 9:00 a.m.) metric values collected between 8:00 a.m. and 9:00 a.m. are forwarded. Subsequently, the metric values received in that interval are sent. So at 9:30 a.m., metric values received between 9:00 a.m. and 9:30 a.m., and at 10:00 a.m. metric values received between 9:30 a.m. and 10:00 a.m., are forwarded.
- Alerts are sent without latency. Each outbound message only has one alert embedded in it.
- Forwarded Service-level Agreement (SLA) data is the SLA value for the selected time period. For a 24-hour scenario, if an outbound session with an SLA metric is scheduled at January 15th at 4:00 p.m., the value forwarded is the SLA value from January 14th at 4:01 pm to January 15th at 4:00 p.m.

3.2 Data Exchange Concepts

The following sections explain the major concepts that you must understand to successfully set up a data exchange environment between Enterprise Manager and an external management system.

3.2.1 Data Exchange Hub

A data exchange hub is a JMS-compliant server that acts as the conduit between Enterprise Manager and an external management system. Data is sent and received between external systems and Cloud Control through such a hub. The hub should be configured with known JMS destination information (Outbound JMS Destinations) so that the messages can be sent and retrieved seamlessly. The Data Exchange Hub page shows a list of existing Data Exchange hubs and their related JNDI Service Provider URLs, provided that at least one hub has already been created. Examples of a hub are WebLogic Server (WLS), Oracle Containers for JEE (OC4J), and so forth.

See Also: "Creating a Data Exchange Hub" on page 3-5

3.2.2 Inbound Data Exchange Session

An inbound data exchange session is created to receive business indicators, events, or both from the data source of an external system to Enterprise Manager.

See Also: "Creating an Inbound Data Exchange Session" on page 3-28

3.2.3 Outbound Data Exchange Session

An outbound data exchange session is created to send metric values, alerts, target availability, or a combination of them from Enterprise Manager to an external system.

The data can be sent in either of the following formats:

- Normalized message format
- Denormalized message format

3.2.3.1 Normalized Message Format

In this format, data is sent in two phases.

- **Session Setup Phase** — Meta information for targets and metrics such as target name, target type, metric name, and metric column are sent along with their GUIDs when the session is created in Enterprise Manager Cloud Control.
- **Session Execution Phase** — Actual metrics are sent when the session is executed. They are tagged with the GUIDs to avoid sending redundant meta information for every message, thereby keeping the wire footprint low.

This message format is effective if the external system is backed by a persistence store, such as a database, so that it can retrieve the metadata by joining the tables when rendering the charts or reports based on GUIDs.

3.2.3.2 Denormalized Message Format

In this format, target and metric meta information is sent along with every message in the session execution phase. No messages are sent during the session setup phase. This message format is effective if the external system is not backed by a persistence store. Though each message repeats the meta information, digesting the data for charting and reporting is easier.

See Also: "Creating an Inbound Data Exchange Session" on page 3-28

3.2.4 Message Schemas

To correctly parse and interpret the contents, it is imperative for the external system to understand the syntax and semantics of the XML messages embedded in the JMS destinations. The schema of the message varies depending on the message format (normalized or denormalized).

The same JMS destinations are used for both formats; therefore, sessions with different message formats should not run concurrently because it confuses the consumer of these messages. Oracle recommends that the sessions with different formats be run exclusively.

3.2.5 Data Source

Data source is a logical representation of an external system source from which business indicators or events are retrieved. A data source definition represents the following:

- The structure and schema of the business content (business indicators) received from the external system.
- The transport (JMS destinations) information by way of which the external data (business events and indicators) is received.
- Associated target in Enterprise Manager to which the external data (business events and indicators) is associated.

3.2.6 Average Data

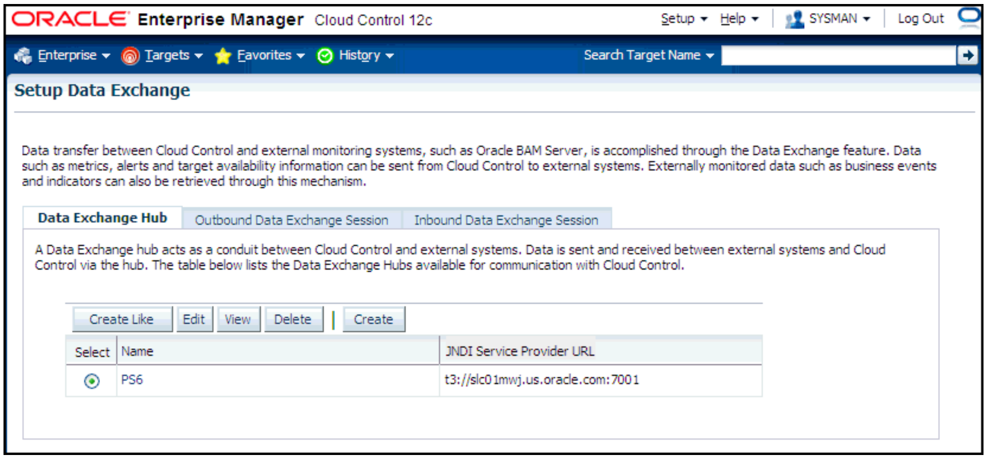
Besides selecting raw metrics, you can also select average metrics for intervals of 24 hours, 7 days, and 31 days. The following conditions apply:

- The default is raw metrics per session.
- You cannot mix and match raw data or different levels of average data per session. For a given session, all data must be raw, one of the average types, or of the same average level.
- Alerts are always sent real-time and have no bearing for the average selection.
- You can also send SLA and target availability using average data, expressed as a percentage.

3.3 Setting up a Data Exchange Connector

1. From Enterprise Manager Cloud Control, click **Setup.**
2. Click **Extensibility**.
3. Click **Data Exchange**.

   The Data Exchange page appears as shown in Figure 3-2.



Setup Data Exchange

Data transfer between Cloud Control and external monitoring systems, such as Oracle BAM Server, is accomplished through the Data Exchange feature. Data such as metrics, alerts and target availability information can be sent from Cloud Control to external systems. Externally monitored data such as business events and indicators can also be retrieved through this mechanism.

A Data Exchange hub acts as a conduit between Cloud Control and external systems. Data is sent and received between external systems and Cloud Control via the hub. The table below lists the Data Exchange Hubs available for communication with Cloud Control.

Select	Name	JNDI Service Provider URL
●	PS6	t3://slc01mwj.us.oracle.com:7001

4. Set up the Data Exchange Connector.

   The following sections provide information required to set up a Data Exchange Connector:

   - Creating a Data Exchange Hub
   - Creating an Outbound Data Exchange Session
   - Creating an Inbound Data Exchange Session

3.3.1 Creating a Data Exchange Hub

The following JMS servers are certified and supported:

- WebLogic Server 8.1 series and above
- OC4J 10.1.3.1 series
- Oracle Enterprise Service Bus (OESB) 10.1.3.1 series
- OC4J 10.1.2.0 series

Note: The use of other third-party JMS-compliant servers may be possible, but such usage is neither certified nor supported. See Section 3.3.2, "Using a Third-Party JMS Server as a Data Exchange Hub" for information on using a third-party JMS server.

Do the following to create a data exchange hub:

1. In the Data Exchange: Data Exchange Hub page click **Create**. The Create Data Exchange Hub page appears.
   a. Specify a unique name for the Data Exchange hub.
   b. Provide the JNDI Service Provider URL for the Data Exchange hub.
   c. Select a context factory name from the list according to the matrix in Table 3–2.

Table 3–2 Context Factory Name

Hub Corresponds to ...	Context Factory
WebLogic Server	weblogic.jndi.WLInitialContextFactory
10.1.2.X OC4J	com.evermind.server.rmi.RMIInitialContextFactory
10.1.3.X OC4J	oracle.j2ee.rmi.RMIInitialContextFactory
Non-WebLogic Server and non-OC4j JMS Server	For third-party servers, select Other and provide a factory name in the Enter JNDI Initial Context Factory Name field.

   d. Provide the user credentials to access this hub.

2. Configure the JMS server with the required JMS topic names and/or queue names.

See Also: "Inbound JMS Destinations" on page 31

3. Click **OK** to save the configuration and return to the Data Exchange: Data Exchange Hub page.

After you create a hub for data exchange, you can set up an outbound or inbound data exchange session.

3.3.2 Using a Third-Party JMS Server as a Data Exchange Hub

Caution: The use of third-party JMS servers is uncertified and unsupported.

Although such usage is uncertified and unsupported, it is possible to use third-party JMS servers as a Data Exchange hub in a development or test scenario by performing the following procedure.

1. Copy your JMS server's JMS client libraries to the following location:

```
$ORACLE_HOME/middleware/oms/sysman/archives/emgc/deployments/
EMGC_DOMAIN/emgc.ear/APP-INF/lib
```

2. Restart Cloud Control after copying the `.jar` file(s).

3. Create the JMS destinations according to the procedures specified for your JMS server. Refer to the list of Topics and Queues on page 3-9.

3.3.3 Creating an Outbound Data Exchange Session

To create an outbound data exchange session, specify the input provided in the following procedure for the respective pages of the setup wizard.

1. From the Data Exchange page, click the **Outbound Data Exchange Session** link.
2. Click **Create**. The Session Setup step of the wizard appears.
   a. Ensure that you have access to at least one Data Exchange hub that is configured with the topics to receive data from Enterprise Manager. To set up a Data Exchange hub, see "Creating a Data Exchange Hub" on page 3-5.

   b. Specify a unique name for this outbound data exchange session in the Name field.

   c. Select a Data Exchange hub from the list of hubs already created and listed alphabetically. By default, the first hub in the list is selected.

   d. In the Users field, enter the name of a BAM user or a comma-separated list of BAM users (for example: user1, user2) on the external system who can access the data being sent. The name or name of these users for this session will be associated with the Oracle BAM server. The receiving system must interpret and enforce security. If you leave this field blank, no user information is sent.

   e. For the Destination Type drop-down, select one of the messaging models to send or receive messages from Cloud Control. Queues provide point-to-point messaging interaction based on Queue logic. Topics provide broadcast model-based subscription methodology.

   If the session uses Topic as the Destination Type, all messages for this session are sent on the corresponding pre-defined topics. For Queues, all messages are sent on the pre-defined Queues. Mix-and-match of destination types for a particular session is not supported.

   f. Select either a **Normalized** or **Denormalized** message format. For more information on these formats, see "Outbound Data Exchange Session" on page 3-3.

   g. In the Metrics Granularity drop-down, select whether you want to send raw metrics data through an outbound session, or whether you want to send average data for 24 hours, 7 days, or 31 days.

   h. Click **Next**. The Select Targets and Metrics/Alerts/Availability step of the wizard appears.
3. Click **Add**. The Add Target page of this wizard step appears.
4. Click the search icon to invoke the Search and Select Targets pop-up, choose a target that is to be part of the Data Exchange transactions being set up, then click **Select**. A list of all metrics and objects for the target now appear on the Add Target page.
5. Specify which metrics and alerts you want to send during the outbound session for the target. Specify which metrics and alerts you want to send during the outbound session for the target. You can use the batch selection buttons if you want to select or deselect all targets and all alerts.
   - You need to select **Send Metric Values** for the Availability metric for target availability information.

- You cannot select the Send Alerts for the Service-Level Agreement (SLA) metric. This metric is only applicable and available for service targets. For example, host targets do not have this metric.
- Some metrics may require you to specify an object. For example, you can qualify the Filesystem Space Available (%) metric by the name of a particular mount point.

6. Click **OK** to save your selections and return to the Select Targets and Metrics/Alerts/Availability step.

   You can now view or edit your selections. You can also use your selections as a template for another target by clicking the **Add-like** icon.

7. Click **Next** if you are satisfied with your metric values and alerts selections. The Schedule step of the wizard appears. Select one of the following scheduling choices:

   - **Schedule Later** — You can defer scheduling and subsequently schedule the session from the Outbound Data Exchange Session sub-page after you click Finish in the Review step of the wizard.
   - **Schedule Now** — Choose one of the following sub-types:
     - **One Time (Immediately)**: If you select this option, the session runs once just when you finish creating it.
     - **One Time (Later):** If you select this option, you need to specify a time zone and a start date and time for the session.
     - Repeating: For this default option, you need to specify the time zone and the start time. Additionally, you can specify the frequency type and interval at which you want the session to run, and whether it should be repeated indefinitely or until a specified time and date.

8. Click **Next** or **Review** to go to the Review step of the wizard.

   If you need to make changes, click **Back** until you reach the step you need to change. Otherwise, go to the next step.

9. Click **Finish**. The Outbound Data Exchange Session sub-page reappears and shows your newly created session and its status in the table.

   Before the job finishes executing, you can either view the schedule by clicking the **View Schedule** link in the Actions column and then stop the execution if desired, or you can stop the execution immediately by clicking **Stop**.

3.3.4 Outbound JMS Destinations

Predefined topic and queue names are used to send data from Enterprise Manager to external systems through the hub. You should configure the data exchange hub with the JMS destination information specified in Table 3–3 through Table 3–9. It is not mandatory to define both topics and queues. If you always want to use the topics, for example, you can remove the queue definitions or not initially create them, and vice versa.

Example - Configuring JMS Destinations for WebLogic Server

You can use any JMS-compliant server with the Data Exchange Connector as described in this example.

To configure the JMS destinations for WebLogic Server, do the following:

1. Use the pre-packaged WLST python scripts available in the `$ORACLE_HOME/sysman/bam` directory.
2. Set the proper CLASSPATH before going to the next step. You can set the CLASSPATH by running `setWLSEnv.sh` found under `ORACLE_HOME` (typically in the `middleware/wlserver_10.3/server/bin` directory).
3. Use `configEMSYSJMSSystemResource.py` found in the directory in step 1 to create the required JMS topics and queues:

   ```
   java weblogic.WLST comfigEMSYSJMSSystemResource.py <jndi provider URL>
   <username> <password> <WLS server name>
   ```

   Example:

   ```
   java weblogic.WLST configEMSYSJMSSystemResource.py "t3://localhost:7001"
   weblogic welcome1 AdminServer
   ```

   A successful run of the script produces the following components:

   - **Resources and Destinations**:
     - EMSYSJMSServer for JMS Server
     - EMSYSJMSSystemResource for JMS System Resource
     - EMSYSJMSServerDeployment for sub-deployment
   - **Connection Factories**:
     - jms/EMSYSTopicConnectionFactory
     - jms/EMSYSQueueConnectionFactory
   - **Topics**:
     - jms/EMSYSTargetsTopic
     - jms/EMSYSMetricsTopic
     - jms/EMSYSAlertsDataTopic
     - jms/EMSYSMetricsDataTopic
     - jms/EMSYSSecurityFilterTopic
     - jms/EMSYSTargetStatusTopic
     - jms/EMSYSTargetSLATopic
     - jms/EMSYSMetricsDataLast24HoursTopic
     - jms/EMSYSMetricsDataLast7DaysTopic
     - jms/EMSYSMetricsDataLast31DaysTopic
     - jms/EMSYSTargetStatusLast24HoursTopic
     - jms/EMSYSTargetStatusLast7DaysTopic
     - jms/EMSYSTargetStatusLast31DaysTopic
     - jms/EMSYSTargetSLALast24HoursTopic
     - jms/EMSYSTargetSLALast7DaysTopic
     - jms/EMSYSTargetSLALast31DaysTopic
   - **Queues**:
     - jms/EMSYSTargetsQueue

- jms/EMSYSMetricsQueue
- jms/EMSYSAlertsDataQueue
- jms/EMSYSMetricsDataQueue
- jms/EMSYSSecurityFilterQueue
- jms/EMSYSTargetStatusQueue
- jms/EMSYSTargetSLAQueue
- jms/EMSYSMetricsDataLast24HoursQueue
- jms/EMSYSMetricsDataLast7DaysQueue
- jms/EMSYSMetricsDataLast31DaysQueue
- jms/EMSYSTargetStatusLast24HoursQueue
- jms/EMSYSTargetStatusLast7DaysQueue
- jms/EMSYSTargetStatusLast31DaysQueue
- jms/EMSYSTargetSLALast24HoursQueue
- jms/EMSYSTargetSLALast7DaysQueue
- jms/EMSYSTargetSLALast31DaysQueue

4. Use `EMSYSJMSSystemResource.py` to remove and clean up the JMS destinations the script `configEMSYSJMSSystemResource.py` created. CLASSPATH should also be set as defined in Step 2 for the following command:

   ```
   java weblogic.WLST deleteEMSYSJMSSystemResource.py <jndi provider URL>
   <username> <password>
   ```

   Example:

   ```
   java weblogic.WLST deleteEMSYSJMSSystemResource.py "t3://localhost:7001"
   weblogic welcome1
   ```

The JMS destinations shown in Table 3–3 are used for an outbound Data Exchange session.

Table 3–3 JMS Destination for Targets

Properties	Description
ConnectionFactory Name	jms/EMSYSTopicConnectionFactory or jms/EMSYSQueueConnectionFactory
Destination Name	jms/EMSYSTargetsTopic or jms/EMSYSTargetsQueue
JMS Message Type	Text message
Description	Target metadata information, such as target name and type are sent on this destination.

Note:

- For outbound sessions using Topic as Destination Type, `jms/EMSYSTopicConnectionFactory` and all topic versions (`ms/EMSYSTargetsTopic` and so forth) are used.
- For Destination Type as Queue, `jms/EMSYSQueueConnectionFactory` and queue versions (`jms/EMSYSTargetsQueue` and so forth) are used.

Table 3–4 JMS Destination for Metrics

Properties	Description
ConnectionFactory Name	jms/EMSYSTopicConnectionFactory or jms/EMSYSQueueConnectionFactory
Destination Name	jms/EMSYSMetricsTopic or jms/EMSYSMetricsQueue
JMS Message Type	Text message
Description	Metric metadata information, such as metric name, column, and target type are sent on this destination.

Table 3–5 JMS Destination for Raw or Average Metric Data

Properties	Description
ConnectionFactory Name	jms/EMSYSTopicConnectionFactory or jms/EMSYSQueueConnectionFactory
Destination Name	■ Raw: jms/EMSYSMetricsDataTopic or jms/EMSYSMetricsDataQueue ■ Last 24 Hours: jms/EMSYSMetricsDataLast24HoursTopic or jms/EMSYSMetricsDataLast24HoursQueue ■ Last 7 Days: jms/EMSYSMetricsDataLast7DaysTopic or jms/EMSYSMetricsDataLast7DaysQueue ■ Last 31 Days: jms/EMSYSMetricsDataLast31DaysTopic or jms/EMSYSMetricsDataLast31DaysQueue
JMS Message Type	Text message
Description	This destination is used to send raw or average metric values.

Table 3–6 JMS Destination for Raw or Average Target Status

Properties	Description
ConnectionFactory Name	jms/EMSYSTopicConnectionFactory or jms/EMSYSQueueConnectionFactory

Table 3–6 (Cont.) JMS Destination for Raw or Average Target Status

Properties	Description
Destination Name	■ Raw: jms/EMSYSTargetStatusTopic or jms/EMSYSTargetStatusQueue ■ Last 24 Hours: jms/EMSYSTargetStatusLast24HoursTopic or jms/EMSYSTargetStatusLast24HoursQueue ■ Last 7 Days: jms/EMSYSTargetStatusLast7DaysTopic or jms/EMSYSTargetStatusLast7DaysQueue ■ Last 31 Days: jms/EMSYSTargetStatusLast31DaysTopic or jms/EMSYSTargetStatusLast31DaysQueue
JMS Message Type	Text message
Description	This destination is used to send raw (numeric value) or average (percentage) target status information.

Table 3–7 JMS Destination for Security Filter

Properties	Description
ConnectionFactory Name	jms/EMSYSTopicConnectionFactory or jms/EMSYSQueueConnectionFactory
Destination Name	jms/EMSYSSecurityFilterTopic
JMS Message Type	Text message
Description	Security filter information is sent on this destination.

Table 3–8 JMS Destination for Alert Data

Properties	Description
ConnectionFactory Name	jms/EMSYSTopicConnectionFactory or jms/EMSYSQueueConnectionFactory
Destination Name	jms/EMSYSAlertsDataTopic
JMS Message Type	Text message
Description	Alerts are sent on this destination.

Table 3–9 JMS Destination for Raw or Average SLA Data

Properties	Description
ConnectionFactory Name	jms/EMSYSTopicConnectionFactory or jms/EMSYSQueueConnectionFactory

Table 3–9 (Cont.) JMS Destination for Raw or Average SLA Data

Properties	Description
Destination Name	■ Raw: jms/EMSYSATargetSLATopic or jms/EMSYSATargetSLAQueue ■ Last 24 Hours: jms/EMSYSATargetSLALast24HoursTopic or jms/EMSYSATargetSLALast24HoursQueue ■ Last 7 Days: jms/EMSYSATargetSLALast7DaysTopic or jms/EMSYSATargetSLALast7DaysQueue ■ Last 31 Days: jms/EMSYSATargetSLALast31DaysTopic or jms/EMSYSATargetSLALast31DaysQueue
JMS Message Type	Text message
Description	This destination is used to send raw or average target SLA data. Raw or average SLA metrics are only shown for Service targets.

3.3.5 Outbound Message Schema

The following sections explain the outbound message schema. The schema varies depending on whether the message format is normalized or denormalized.

To avoid regressions and conflicts between different type of metric data, the XML element and target names differ for raw versus historical data. Table 3–10 shows the differences for these based on the type of granularity generated.

Table 3–10 Raw Versus Historical Data

Granularity	Metric Element Name	Target Status Element Name	SLA Target Name
Raw	MetricData	TargetStatus	TargetSLA
Last 24 Hours	MetricDataLast24Hours	TargetStatusLast24Hours	TargetSLALast24Hours
Last 7 Days	MetricDataLast7Days	TargetStatusLast7Days	TargetSLALast7Days
Last 31 Days	MetricDataLast31Days	TargetStatusLast31Days	TargetSLALast31Days

3.3.5.1 Normalized Message Format

The schema for outgoing messages for a normalized format is as follows:

Normalized Target Message

For each selected target, corresponding target metadata information is sent to the external system during the session setup phase. The schema of these messages is as follows:

Table 3–11 Normalized Target Message

Elements and Sample	Description
Path Expression	EMSYSData/Target
Schema File Location	$ORACLE_HOME/sysman/bam/OutboundTarget.xsd

Table 3–11 (Cont.) Normalized Target Message

<table>
<tr><th>Elements and Sample</th><th>Description</th></tr>
<tr><td>Destination</td><td>jms/EMSYSTargetsTopic or jms/EMSYSTargetsQueue</td></tr>
<tr><td>Schema</td><td>

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://xmlns.oracle.com/EnterpriseManager/DataExch
ange/10203/OutboundData/"
xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10
203/OutboundData/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
     <xs:element name="EMSYSData" type="de:EMSYSDataType"/>
     <!-- Define the root element -->
     <xs:complexType name="EMSYSDataType">
          <xs:sequence>
               <xs:element name="Target" type="de:TargetType"
minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
     </xs:complexType>
     <!-- Define the Target Type -->
     <xs:complexType name="TargetType">
          <xs:all>
               <xs:element name="TargetName" type="xs:string"/>
               <xs:element name="TargetType" type="xs:string"/>
               <xs:element name="TargetGUID" type="xs:string"/>
          </xs:all>
     </xs:complexType>
</xs:schema>
```

</td></tr>
<tr><td>Sample</td><td>

```
<de:EMSYSData
xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10
203/OutboundData/">
   <Target>
      <TargetName>/ade/foo_core9/oracle.stacd16.us.oracle.com_
home</TargetName>
     <TargetType>oc4j</TargetType>
     <TargetGUID>852464ac3e4176460458297faaffe926</TargetGUID>
  </Target>
  <Target>
     <TargetName>stacd16.us.oracle.com</TargetName>
     <TargetType>host</TargetType>
      <TargetGUID>00645a665bfd9b72b2a6bb6ef49606b0</TargetGUID>
   </Target>
</de:EMSYSData>
```

</td></tr>
</table>

Normalized Metric Message

For each selected metric, corresponding metric metadata information is sent to the external system during the session setup. The schema of these messages is as follows:

Table 3–12 Normalized Metric Message

Elements and Sample	Description
Path Expression	EMSYSData/Metric

Table 3–12 (Cont.) Normalized Metric Message

<table>
<thead>
<tr><th>Elements and Sample</th><th>Description</th></tr>
</thead>
<tbody>
<tr><td>Schema File Location</td><td>$ORACLE_HOME/sysman/bam/OutboundSecurityFilter.xsd</td></tr>
<tr><td>Destination</td><td>jms/EMSYSMetricsTopic or jms/EMSYSMetricsQueue</td></tr>
<tr><td>Schema</td><td>

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://xmlns.oracle.com/EnterpriseManager/DataExchan
ge/10203/OutboundData/"
xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/1020
3/OutboundData/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
          <xs:element name="EMSYSData" type="de:EMSYSDataType"/>
     <!-- Define the root element -->
     <xs:complexType name="EMSYSDataType">
          <xs:sequence>
               <xs:element name="Metric" type="de:MetricType"
minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
     </xs:complexType>
     <!-- Define the Metric Type -->
     <xs:complexType name="MetricType">
          <xs:all>
               <xs:element name="MetricName" type="xs:string"/>
               <xs:element name="MetricType" type="xs:string"/>
               <xs:element name="MetricGUID" type="xs:string"/>
               <xs:element name="MetricColumn" type="xs:string"
minOccurs="0"/>
          </xs:all>
     </xs:complexType>
</xs:schema>
```

</td></tr>
<tr><td>Sample</td><td>

```
<de:EMSYSData
xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/1020
3/OutboundData/">
   <Metric>
    <MetricColumn>DiskActivityavserv</MetricColumn>
    <MetricGUID>6ca028d5078fe542b2ee0c1b013727d7</MetricGUID>
    <TargetType>host</TargetType>
    <MetricName>DiskActivity</MetricName>
  </Metric>
  <Metric>
    <MetricColumn>cpuUtil</MetricColumn>
    <MetricGUID>0c71a1afac2d7199013837da35522c08</MetricGUID>
    <TargetType>host</TargetType>
    <MetricName>Load</MetricName>
   </Metric>
</de:EMSYSData>
```

</td></tr>
</tbody>
</table>

Normalized Security Filter Message

External systems that consume data from Enterprise Manager can enforce access control based on the session name. This can be achieved by capturing the security filter. The schema of these security filter messages is as follows:

Table 3–13 Normalized Security Filter Message

Elements and Sample	Description
Path Expression	EMSYSData/SecurityFilter
Schema File Location	$ORACLE_HOME/sysman/bam/OutboundMetric.xsd
Destination	jms/EMSYSSecurityFilterTopic or jms/EMSYSSecurityFilterQueue

Schema	`<?xml version="1.0" encoding="UTF-8"?>`
	`<xs:schema`
	`targetNamespace="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203/OutboundData/"`
	`xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203/OutboundData/" xmlns:xs="http://www.w3.org/2001/XMLSchema"`
	`elementFormDefault="qualified" attributeFormDefault="unqualified">`
	`     <xs:element name="EMSYSData" type="de:EMSYSDataType"/>`
	`     <!-- Define the root element -->`
	`     <xs:complexType name="EMSYSDataType">`
	`          <xs:sequence>`
	`               <xs:element name="SecurityFilter"`
	`type="de:SecurityFilterType" minOccurs="0" maxOccurs="unbounded"/>`
	`          </xs:sequence>`
	`     </xs:complexType>`
	`     <!-- Define the Security Filter Type -->`
	`     <xs:complexType name="SecurityFilterType">`
	`          <xs:all>`
	`               <xs:element name="SessionName" type="xs:string"/>`
	`               <xs:element name="UserName" type="xs:string"/>`
	`          </xs:all>`
	`     </xs:complexType>`
	`</xs:schema>`

Sample	`<de:EMSYSData`
	`xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203/OutboundData/">`
	`  <SecurityFilter>`
	`    <SessionName>LoanSession</SessionName>`
	`    <UserName>LoanAdminUser</UserName>`
	`  </SecurityFilter>`
	`</de:EMSYSData>`

Normalized Metric Data Message

In the normalized message format, the metrics are sent along with the GUIDs to avoid sending meta information for every message. The schema of this metric message is as follows:

Table 3–14 Normalized Metric Data Message

Elements and Sample	Description
Path Expression	EMSYSData/MetricData
Schema File Location	$ORACLE_HOME/sysman/bam/ OutboundNormalizedMetricsData.xsd
Destination	jms/EMSYSMetricsDataTopic or jms/EMSYSMetricsDataQueue

Table 3–14 (Cont.) Normalized Metric Data Message

Elements and Sample	Description
Schema	<?xml version="1.0" encoding="UTF-8"?> <xs:schema targetNamespace="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203/OutboundData/" xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203/OutboundData/" xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified"> <xs:element name="EMSYSData" type="de:EMSYSDataType"/> <!-- Define the root element --> <xs:complexType name="EMSYSDataType"> <xs:sequence> <xs:element name="MetricData" type="de:MetricDataType" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> <!-- Define the Metric Data Type --> <xs:complexType name="MetricDataType"> <xs:all> <xs:element name="SessionName" type="xs:string"/> <xs:element name="TargetGUID" type="xs:string"/> <xs:element name="MetricGUID" type="xs:string"/> <xs:element name="Timestamp" type="xs:dateTime"/> <xs:element name="Value" type="xs:float" minOccurs="0"/> <xs:element name="StringValue" type="xs:string" minOccurs="0"/> <xs:element name="KeyValue" type="xs:string" minOccurs="0"/> </xs:all> </xs:complexType> </xs:schema>
Sample	<de:EMSYSData xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203/OutboundData/"> <MetricDataLast24Hrs> <MetricGUID>df8067b7d515747434b58d69230b8451</MetricGUID> <Value>0.67</Value> <Timestamp>07/13/2012 14:44:42</Timestamp> <SessionName>Session1</SessionName> <TargetGUID>00645a665bfd9b72b2a6bb6ef49606b0</TargetGUID> </MetricDataLast24Hrs> </de: EMSYSData >

Normalized Alert Message

In the normalized message format, the alerts are sent along with the GUIDs to avoid sending meta information for every message. The schema of this alert message is as follows:

Table 3–15 Normalized Alert Message

Elements and Sample	Description
Path Expression	EMSYSData/Alert
Schema File Location	$ORACLE_HOME/sysman/bam/ OutboundNormalizedAlertsData.xsd
Destination	jms/EMSYSAlertsDataTopic or jms/EMSYSAlertsDataQueue

Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://xmlns.oracle.com/EnterpriseManager/DataExchan
ge/10203/OutboundData/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/1020
3/OutboundData/" elementFormDefault="qualified"
attributeFormDefault="unqualified">
     <xs:element name="EMSYSData" type="de:EMSYSDataType"/>
     <!-- Define the root element -->
     <xs:complexType name="EMSYSDataType">
          <xs:sequence>
               <xs:element name="Alert" type="de:AlertType"
minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
     </xs:complexType>
     <!-- Define the Alert Type -->
     <xs:complexType name="AlertType">
          <xs:all>
               <xs:element name="SessionName" type="xs:string"/>
               <xs:element name="TargetGUID" type="xs:string"/>
               <xs:element name="MetricGUID" type="xs:string"/>
               <xs:element name="Timestamp" type="xs:dateTime"/>
               <xs:element name="Severity" type="xs:string"/>
               <xs:element name="Value" type="xs:float"
minOccurs="0"/>
               <xs:element name="Message" type="xs:string"
minOccurs="0"/>
               <xs:element name="KeyValue" type="xs:string"
minOccurs="0"/>
          </xs:all>
     </xs:complexType>
</xs:schema>
```

Sample

```
<de:EMSYSData
xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/
/10203/OutboundData/">
   <Alert>
    <MetricGUID>0c71a1afac2d7199013837da35522c08</MetricGUID>
    <Value>25.35</Value>
    <Message>CPU Utilization is 25.35%, crossed warning (15) or
critical (95) threshold.</Message>
    <Severity>Warning</Severity>
               <Timestamp>07/13/2012 14:59:42</Timestamp>
    <SessionName>Session9</SessionName>
    <TargetGUID>00645a665bfd9b72b2a6bb6ef49606b0</TargetGUID>
   </Alert>
</de:EMSYSData
```

List of Severities

- CLEAR
- INFO
- WARNING
- CRITICAL
- AGENT UNREACHABLE CLEAR
- AGENT UNREACHABLE START
- BLACKOUT END
- BLACKOUT START
- METRIC ERROR END
- METRIC ERROR START

Normalized Target Availability Message

The schema of a normalized target availability information message is as follows:

Table 3–16 Normalized Target Availability Message

Elements and Sample	Description
Path Expression	EMSYSData/TargetStatus
Schema File Location	$ORACLE_HOME/sysman/bam/OutboundNormalizedTargetStatus.xsd
Destination	jms/EMSYSTargetStatusTopic or jms/EMSYSTargetStatusQueue

Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://xmlns.oracle.com/EnterpriseManager/DataExchang
e/10203/OutboundData/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203
/OutboundData/" elementFormDefault="qualified"
attributeFormDefault="unqualified">
     <xs:element name="EMSYSData" type="de:EMSYSDataType"/>
     <!-- Define the root element -->
     <xs:complexType name="EMSYSDataType">
          <xs:sequence>
               <xs:element name="TargetStatus"
type="de:TargetsStatusType" minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
     </xs:complexType>
     <!-- Define the Target Status Type -->
     <xs:complexType name="TargetStatusType">
          <xs:all>
               <xs:element name="SessionName" type="xs:string"/>
               <xs:element name="TargetGUID" type="xs:string"/>
               <xs:element name="Status" type="xs:integer"/>
               <xs:element name="Timestamp" type="xs:dateTime"/>
          </xs:all>
     </xs:complexType>
</xs:schema>
```

Table 3–16 (Cont.) Normalized Target Availability Message

Elements and Sample	Description

Sample	`<de:EMSYSData xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203/OutboundData/">`
	`<TargetStatusLast7Days>`
	`<Status>1</Status>`
	`<Timestamp>07/11/2012 16:21:53</Timestamp>`
	`<SessionName>Session1</SessionName>`
	`<TargetGUID>00645a665bfd9b72b2a6bb6ef49606b0</TargetGUID>`
	`</TargetStatusLast7Days>`
	`</de:EMSYSData>`

Possible status values are provided in the following table:

Value	Status
1	Target is up and reachable
-1	■ Target is down ■ Metric error ■ Agent is down
0	■ Blackout ■ Target is not monitored ■ Target is unknown

Normalized Target SLA Message

The schema of a normalized target SLA information message is as follows:

Table 3–17 Normalized Target SLA Message

Elements and Sample	Description
Path Expression	EMSYSData/TargetSLA
Schema File Location	$ORACLE_HOME/sysman/bam/OutboundNormalizedTargetSLA.xsd
Destination	jms/EMSYSTargetSLATopic or jms/EMSYSTargetSLAQueue

Table 3–17 (Cont.) Normalized Target SLA Message

Elements and Sample	Description
Schema	

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Schema for Normalized Outbound Target Status message -->
<xs:schema targetNamespace="http://xmlns.oracle.com/EnterpriseManager/
DataExchange/10203/OutboundData/"xmlns:de="http://xmlns.oracle.com/
EnterpriseManager/DataExchange/10203/OutboundData/
" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
      <!-- Define the root element -->
        <xs:element name="EMSYSData" type="de:EMSYSDataType"/>
        <!-- Define the root type -->
        <xs:complexType name="EMSYSDataType">
              <!-- Zero or more TargetSLA elements -->
                <xs:sequence>
                        <xs:element name="TargetSLA"
             type="de:TargetSLAType" minOccurs="0"
             maxOccurs="unbounded"/>
                </xs:sequence>
        </xs:complexType>
        <!-- Define the TargetSLA Type -->
        <xs:complexType name="MetricDataType">
                <xs:all>
                        <xs:element name="SessionName"
                        type="xs:string"/>
                        <xs:element name="TargetGUID"
                        type="xs:string"/>
                        <xs:element name="SLA" type="xs:integer"/>
                        <xs:element name="Timestamp"
                        type="xs:dateTime"/>
                </xs:all>
        </xs:complexType>
</xs:schema>
```

Table 3–17 (Cont.) Normalized Target SLA Message

Elements and Sample	Description

Sample	`<?xml version="1.0" encoding="UTF-8"?>`
	`<!-- Schema for Normalized Outbound Target Status message -->`
	`<xs:schema targetNamespace="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203/OutboundData/"xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203/OutboundData/" xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">`
	`      <!-- Define the root element -->`
	`        <xs:element name="EMSYSData" type="de:EMSYSDataType"/>`
	`        <!-- Define the root type -->`
	`        <xs:complexType name="EMSYSDataType">`
	`              <!-- Zero or more TargetSLA elements -->`
	`                <xs:sequence>`
	`                        <xs:element name="TargetSLA"`
	`             type="de:TargetSLAType" minOccurs="0"`
	`             maxOccurs="unbounded"/>`
	`                </xs:sequence>`
	`        </xs:complexType>`
	`        <!-- Define the TargetSLA Type -->`
	`        <xs:complexType name="MetricDataType">`
	`                <xs:all>`
	`                        <xs:element name="SessionName"`
	`                        type="xs:string"/>`
	`                        <xs:element name="TargetGUID"`
	`                        type="xs:string"/>`
	`                        <xs:element name="SLA" type="xs:integer"/>`
	`                        <xs:element name="Timestamp"`
	`                        type="xs:dateTime"/>`
	`                </xs:all>`
	`        </xs:complexType>`
	`</xs:schema>`
	`<de:EMSYSData`
	`xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203/OutboundData/">`
	` <TargetStatusLast7Days>`
	`    <Status>1</Status>`
	`    <Timestamp>07/11/2012 16:21:53</Timestamp>`
	`    <SessionName>Session1</SessionName>`
	`    <TargetGUID>00645a665bfd9b72b2a6bb6ef49606b0</TargetGUID>`
	` </TargetStatusLast7Days>`
	`</de:EMSYSData>`

3.3.5.2 Denormalized Message Format

Following sections describe the schema for the outgoing messages for denormalized format.

Denormalized Metric Data Message

Schema of a denormalized metric data message is as follows:

Table 3–18 Denormalized Metric Data Message

Elements and Sample	Description
Path Expression	EMSYSData/MetricData
Schema File Location	$ORACLE_HOME/sysman/bam/ OutboundDenormalizedMetricsData.xsd
Destination	jms/EMSYSMetricsDataTopic or jms/EMSYSMetricsDataQueue
Schema	<?xml version="1.0" encoding="UTF-8"?> <xs:schema targetNamespace="http://xmlns.oracle.com/EnterpriseManager/DataExchan ge/10203/OutboundData/" xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/1020 3/OutboundData/" xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified"> <xs:element name="EMSYSData" type="de:EMSYSDataType"/> <!-- Define the root element --> <xs:complexType name="EMSYSDataType"> <xs:sequence> <xs:element name="MetricData" type="de:MetricDataType" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> <!-- Define the Metric Data Type --> <xs:complexType name="MetricDataType"> <xs:all> <xs:element name="SessionName" type="xs:string"/> <xs:element name="TargetName" type="xs:string"/> <xs:element name="TargetType" type="xs:string"/> <xs:element name="MetricName" type="xs:string"/> <xs:element name="MetricColumn" type="xs:string" minOccurs="0"/> <xs:element name="Timestamp" type="xs:dateTime"/> <xs:element name="Value" type="xs:float" minOccurs="0"/> <xs:element name="StringValue" type="xs:string" minOccurs="0"/> <xs:element name="KeyValue" type="xs:string" minOccurs="0"/> </xs:all> </xs:complexType> </xs:schema>
Sample	<de:EMSYSData xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/1020 3/OutboundData/"> <MetricDataLast24Hours> <SessionName>Session1</SessionName> <TargeName>OC4J 10.1.3</TargetName> <TargetType>generic_service</TargetType> <MetricName>Usage Value</MetricName> <Timestamp>2001-12-17T09:30:47-05:00</Timestamp> <Value>3.14159</Value> </MetricDataLast24Hours> </EMSYSData>

Denormalized Alert Message

Schema of a denormalized alert message is as follows:

Table 3–19 Denormalized Alert Message

<table>
<tr><th>Elements and Sample</th><th>Description</th></tr>
<tr><td>Path Expression</td><td>EMSYSData/Alert</td></tr>
<tr><td>Schema File Location</td><td>$ORACLE_HOME/sysman/bam/ OutboundDenormalizedAlertsData.xsd</td></tr>
<tr><td>Destination</td><td>jms/EMSYSAlertsDataTopic or jms/EMSYSAlertsDataQueue</td></tr>
<tr><td>Schema</td><td>

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://xmlns.oracle.com/EnterpriseManager/DataExchang
e/10203/OutboundData/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203
/OutboundData/" elementFormDefault="qualified"
attributeFormDefault="unqualified">
     <xs:element name="EMSYSData" type="de:EMSYSDataType"/>
     <!-- Define the root element -->
     <xs:complexType name="EMSYSDataType">
          <xs:sequence>
               <xs:element name="Alert" type="de:AlertType"
minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
     </xs:complexType>
     <!-- Define the Alert Type -->
     <xs:complexType name="AlertType">
          <xs:all>
               <xs:element name="SessionName" type="xs:string"/>
               <xs:element name="TargetName" type="xs:string"/>
               <xs:element name="TargetType" type="xs:string"/>
               <xs:element name="MetricName" type="xs:string"/>
               <xs:element name="MetricColumn" type="xs:string"
minOccurs="0"/>
               <xs:element name="Timestamp" type="xs:dateTime"/>
               <xs:element name="Severity" type="xs:string"/>
               <xs:element name="Value" type="xs:float"
minOccurs="0"/>
               <xs:element name="Message" type="xs:string"
minOccurs="0"/>
               <xs:element name="KeyValue" type="xs:string"
minOccurs="0"/>
          </xs:all>
     </xs:complexType>
</xs:schema>
```

</td></tr>
</table>

Table 3–19 (Cont.) Denormalized Alert Message

Elements and Sample	Description

Sample	`<de:EMSYSData xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203/OutboundData/">`
	`<Alert>`
	`<SessionName>Session9</SessionName>`
	`<TargeName>OC4J 10.1.3</TargetName>`
	`<TargetType>generic_service</TargetType>`
	`<MetricName>Usage Value</MetricName>`
	`<Value>25.35</Value>`
	`<Message>CPU Utilization is 25.35%, crossed warning (15) or critical (95) threshold.</Message>`
	`<Severity>Warning</Severity>`
	`<Timestamp>07/13/2012 14:59:42</Timestamp>`
	`</Alert>`
	`</de:EMSYSData>`

Denormalized Target Availability Message

Schema of a denormalized target availability message is as follows:

Table 3–20 Denormalized Target Availability Message

Elements and Sample	Description
Path Expression	EMSYSData/TargetStatus
Schema File Location	$ORACLE_HOME/sysman/bam/ OutboundDenormalizedTargetStatus.xsd
Destination	jms/EMSYSTargetStatusTopic or jms/EMSYSTargetStatusQueue

Table 3–20 (Cont.) Denormalized Target Availability Message

Elements and Sample	Description

Schema	`<?xml version="1.0" encoding="UTF-8"?>`
	`<xs:schema`
	`targetNamespace="http://xmlns.oracle.com/EnterpriseManager/DataExchang`
	`e/10203/OutboundData/" xmlns:xs="http://www.w3.org/2001/XMLSchema"`
	`xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203`
	`/OutboundData/" elementFormDefault="qualified"`
	`attributeFormDefault="unqualified">`
	`     <xs:element name="EMSYSData" type="de:EMSYSDataType"/>`
	`     <!-- Define the root element -->`
	`     <xs:complexType name="EMSYSDataType">`
	`          <xs:sequence>`
	`               <xs:element name="TargetStatus"`
	`type="de:TargetStatusType" minOccurs="0" maxOccurs="unbounded"/>`
	`          </xs:sequence>`
	`     </xs:complexType>`
	`     <!-- Define the Target Status Type -->`
	`     <xs:complexType name="TargetStatusType">`
	`          <xs:all>`
	`               <xs:element name="SessionName" type="xs:string"/>`
	`               <xs:element name="TargetName" type="xs:string"/>`
	`               <xs:element name="TargetType" type="xs:string"/>`
	`               <xs:element name="Status" type="xs:integer"/>`
	`               <xs:element name="Timestamp" type="xs:dateTime"/>`
	`          </xs:all>`
	`       </xs:complexType>`
	`</xs:schema>`

Sample	`<de:EMSYSData`
	`xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203`
	`/OutboundData/">`
	`  <TargetStatus>`
	`    <Status>1</Status>`
	`    <Timestamp>07/11/2012 16:21:53</Timestamp>`
	`    <SessionName>Session1</SessionName>`
	`     <TargeName>OC4J 10.1.3</TargetName>`
	
	`     <TargetType>generic_service</TargetType>`
	`   </TargetStatus>`
	`</de:EMSYSData>`

Denormalized Target SLA Message

The schema of a denormalized target SLA information message is as follows:

Table 3–21 Denormalized Target SLA Message

Elements and Sample	Description
Path Expression	EMSYSData/TargetSLA
Schema File Location	$ORACLE_HOME/sysman/bam/OutboundDenormalizedTargetSLA.xsd
Destination	jms/EMSYSTargetSLATopic or jms/EMSYSTargetSLAQueue

Table 3–21 (Cont.) Denormalized Target SLA Message

<table>
<tr><th>Elements and Sample</th><th>Description</th></tr>
<tr><td>Schema</td><td><pre>
<?xml version="1.0" encoding="UTF-8"?>
<!-- Schema for Denormalized Outbound Target Status message -->
<xs:schema targetNamespace="http://xmlns.oracle.com/EnterpriseManager/
DataExchange/10203/OutboundData/"
xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/
10203/OutboundData/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
      <!-- Define the root element -->
        <xs:element name="EMSYSData" type="de:EMSYSDataType"/>
        <!-- Define the root type -->
        <xs:complexType name="EMSYSDataType">
               <!-- zero or more target status elements -->
                <xs:sequence>
                        <xs:element name="TargetSLA"
                         type="de:MetricDataType"
                         minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
        </xs:complexType>
        <!-- Define the Target Status Type -->
        <xs:complexType name="TargetSLAType">
                <xs:all>
                        <xs:element name="SessionName"
                         type="xs:string"/>
                        <xs:element name="TargetName"
                         type="xs:string"/>
                        <xs:element name="TargetType"
                         type="xs:string"/>
                        <xs:element name="SLA" type="xs:integer"/>
                        <xs:element name="Timestamp"
                         type="xs:dateTime"/>
                </xs:all>
        </xs:complexType>
</xs:schema>
</pre></td></tr>
<tr><td>Sample</td><td><pre>
<de:EMSYSData
xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203
/OutboundData/">
  <TargetSLA>
    <SLA>100</SLA>
    <Timestamp>07/11/2012 16:21:53</Timestamp>
    <SessionName>Session1</SessionName>
     <TargeName>OC4J 10.1.3</TargetName>
     <TargetType>generic_service</TargetType>
   </TargetSLA>
</de:EMSYSData>
</pre></td></tr>
</table>

3.3.6 Tuning Outbound Session Message Parameters

You can tune outbound session message parameters using the emctl command, as shown in Table 3–22.

> **Note:** For the parameters to become effective, you need to restart OMS after setting the properties.

Table 3–22 Tuneable Outbound Session Message Parameters

Property Name	Default Value	Semantics
oracle.sysman.core.dataExchange.MaxDataPointsPerMessage	100	Number of metric data points within a message.
oracle.sysman.core.dataExchange.IntervalBetweenMessage	2 seconds	Time gap between subsequent JMS messages in seconds.
oracle.sysman.core.dataExchange.FirstDatasetWindow	60 minutes	When sending the first message, date for the past first set data window is sent. The unit is in minutes.

Example 3–1 Command Syntax for Tuning Outbound Session Message Parameters

```
emctl {set property|get property}
 {oracle.sysman.core.dataExchange.MaxDataPointsPerMessage |
  oracle.sysman.core.dataExchange.IntervalBetweenMessage |
  oracle.sysman.core.dataExchange.FirstDatasetWindow}
```

You need to restart OMS after the properties have been set to be effective.

3.3.7 Creating an Inbound Data Exchange Session

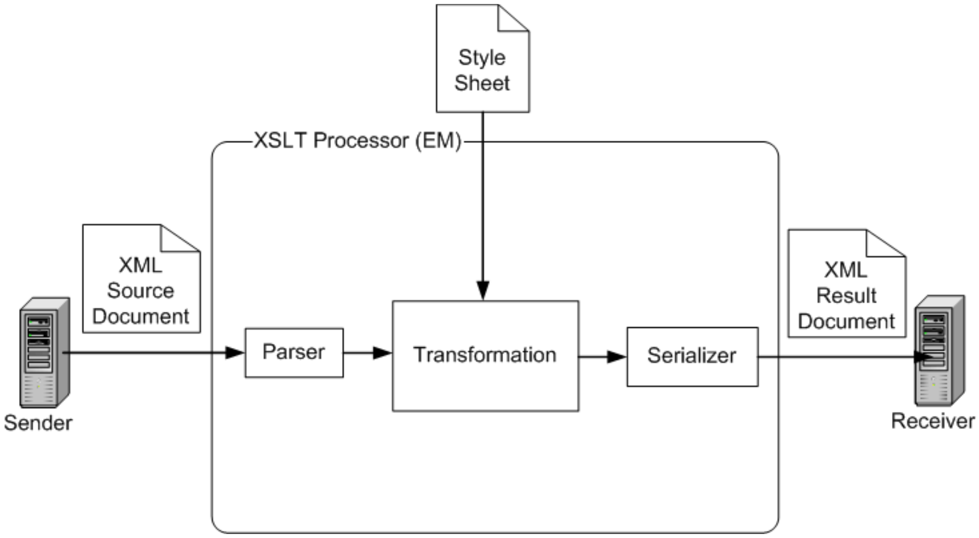
To create an inbound data exchange session, specify the input provided in the following procedure for the respective pages of the setup wizard.

1. From the Data Exchange page, click the **Inbound Data Exchange Session** link.
2. Click **Create**. The Session Setup step of the wizard appears.
   a. Ensure that you have access to at least one Data Exchange hub that is configured with the topics to receive data from Enterprise Manager. To set up a Data Exchange hub, see "Creating a Data Exchange Hub" on page 3-5.
   b. Specify a unique name for this inbound data exchange session in the Name field.
   c. Select a Data Exchange hub from the list of hubs already created and listed alphabetically. By default, the first hub in the list is selected.

      Incoming business events are associated with a corresponding business KPI if you import the KPI to Cloud Control. If not, the event is associated with the special built-in metric, named **ExternalAlertMetric**.
   d. Click **Next**. The Select Business Events/Indicators step of the wizard appears.
3. If you need to send business events to Cloud Control, click **Add Business Events**. Otherwise, skip to step 4. The Select Business Events/Indicators: Add Business Events page appears.
   a. Specify the name of an existing data source from which the event will be retrieved. The data source names within a session should be unique.
   b. Optionally check **Apply XSL Transformation for incoming messages** to apply an XSL style sheet to the incoming message, the conversion of which is shown in Figure 3–2. A multi-line text box appears when you click **Show**, where you can insert an XSL document. If you choose this option, click **Check Syntax** to check the accuracy of your entry before proceeding.

The most common usage for XSLT conversion involves incoming messages transporting business KPIs or events. When the KPIs or events do not produce the expected message (schema), you can apply the XSL at the Cloud Control end rather than changing the message format itself.

Figure 3–2 XSLT Conversion of XML Source Document



c. Specify JMS destination details for incoming events. You need to specify the ConnectionFactory, the destination from which data is retrieved, and an optional Durable Subscriber Name (only needed for topics, not for queues) so that all messages pertaining to the topic go to the specified subscriber. Specifying a Durable Subscriber for topics prevents you from losing any incoming events.

To ensure that an authenticated connection will be created between Cloud Control and the Data Exchange hub, you can specify a user name and password so that the connection can be established with these credentials. Click **Test Connection** to verify that your input is valid.

d. Associate the business events with a target that Cloud Control is monitoring. You can associate business events with any Enterprise Manager monitored target.

The target drop-down lists all the available target types, with the Generic Service target type being the default. If you want to choose the business event associated with a specific target that is not available in the list, add the target from the Targets page and then restart the procedure of creating an inbound session.

e. Click **OK** to save your configuration, then view your input in a tabular format and edit your selections if necessary.

You can also use your selections as a template for another target by clicking **Add-like**.

4. If you need to send business indicators to Cloud Control, click **Add Business Indicators**. Otherwise, skip to step 5. The Select Business Events/Indicators: Add Business Indicators page appears.

a. Specify the name of an existing external data source from which the indicators will be retrieved. The data source names within a session should be unique. If the XML message sent from the data source is namespace-enabled, select the check-box indicating this, and also specify the fully-qualified namespace.

b. Optionally check **Apply XSL Transformation for incoming messages** to apply an XSL style sheet to the incoming message. A multi-line text box appears after you click **Show**, where you can insert an XSL document. If you choose this option, click **Check Syntax** to check the accuracy of your entry before proceeding.

c. Specify the business indicators that need to be sent to Cloud Control by clicking **Add Indicator**. The corresponding metric name for the business indicator is `<Source Name>_<Indicator Name>`. All indicators can only have numeric values.

d. Specify JMS destination details for incoming indicators. You need to specify the ConnectionFactory, the destination from which data is retrieved, and an optional Durable Subscriber Name (only needed for topics, not for queues) so that all messages pertaining to the topic go to the specified subscriber. Specifying a Durable Subscriber topic prevents you from losing any incoming indicators.

   To ensure that an authenticated connection will be created between Cloud Control and the Data Exchange hub, you can specify a user name and password so that the connection can be established with these credentials. Click **Test Connection** to verify that your input is valid.

e. Associate the business indicators with a target that Cloud Control is monitoring. Unlike business events, which can be associated with any target type instance, business indicators can be associated only with instances that are of the Service target type.

   The target drop-downs list all the available target types, with the Generic Service target type being the default. If you want to choose the business event associated with a specific target that is not available in the list, add the target from the Targets page and then restart the procedure of creating an inbound session.

f. Click **OK** to save your input, then view your input in a tabular format and edit your selections if necessary.

   You can also use your selections as a template for another target by clicking **Add-like**.

5. Click **Next** if you are satisfied with the configuration. The Schedule step of the wizard appears. Select one of the following scheduling choices:

   - **Schedule Later** — You can defer scheduling and subsequently schedule the session from the Outbound Data Exchange Session sub-page after you click Finish in the Review step of the wizard.

   - **Schedule Now** — Choose one of the following sub-types:

     – **One Time (Immediately)**: If you select this option, the session runs once just when you finish creating it.

     – **One Time (Later):** If you select this option, you need to specify a time zone and a start date and time for the session.

     – Repeating: For this default option, you need to specify the time zone and the start time. Additionally, you can specify the frequency type and interval at which you want the session to run, and whether it should be repeated indefinitely or until a specified time and date.

6. Click **Next** or **Review** to go to the Review step of the wizard.

   If you need to make changes, click **Back** until you reach the step you need to change. Otherwise, go to the next step.

7. Click **Finish**. The Inbound Data Exchange Session sub-page reappears and shows your newly created session and its status in the table.

   Before the job finishes executing, you can either view the schedule by clicking the **View Schedule** link in the Actions column and then stop the execution if desired, or you can stop the execution immediately by clicking **Stop**.

3.3.8 Inbound JMS Destinations

Unlike the outbound data exchange setup wherein pre-defined topics and queues are used to send Enterprise Manager data, no pre-defined topics or queues are used to receive business performance indicators and events.

However, you should configure the JMS topics or queues used for the data sources in the JMS server used for inbound data exchange session.

3.3.9 Inbound Message Schemas

The following sections define the inbound message schemas. Samples messages are provided along with each schema.

3.3.9.1 Inbound Indicators Schema

After creating the session, the sender can forward the data in XML format using the data exchange hub through the JMS destinations defined in the inbound data exchange session.

Messages can be either namespace qualified or unqualified. If the messages are namespace qualified, the namespace should be entered during the data source setup time.

Qualified XML Message Sample

```
<po:PurchaseOrder xmlns:po:"http://acme.com/Orders">
  <OrderAmount>5000</OrderAmount>
  <NoOfItems> 15 </NoOfItems>
</po:PurchaseOrder>
```

Unqualified XML Message Sample

```
<PurchaseOrder>
   <OrderAmount>5000</OrderAmount>
  <NoOfItems> 15 </NoOfItems>
</PurchaseOrder>
```

3.3.9.2 Message Semantics

The incoming messages should follow the semantics provided below:

- The local name of the top-level element should be same as the data source name as in Example 3–2.
- If the message is qualified, the namespace should be defined during the data source setup time.

- One or more indicators can be sent as child elements within this element as in Example 3–2.
- A sub-element with the Timestamp as the name has special semantics. If a sub-element with the Timestamp name exists, the indicators are inserted with this Timestamp value. If no Timestamp element exists, the current time is used when inserting the indicator into the repository.

  For example, if the request is received as follows with the Timestamp sub-element, the indicators are inserted with this timestamp (2012-09-30 17:43:19.474):

  ```
  <po: PurchaseOrder xmlns:po:"http://acme.com/Orders">
    <OrderAmount>5000</OrderAmount>
    <NoOfItems> 15 </NoOfItems>
    <Timestamp>2012-10-30 17:43:19.474</Timestamp>
  </po: PurchaseOrder>
  ```

  If no Timestamp sub-element is present, the indicators are inserted to the repository with the current timestamp at which they are received.

Example 3–2 Data Source Scenario

You create a Data Source for the incoming business indicators with the Data Source name `Order`. You add the following three KPIs:

- `OrderAmount`
- `NoOfItems`
- `Credit`

  In this case, the incoming XML message should be in the following format:

  ```
  <Order>
     <OrderAmount>35</OrderAmount>
     <NoOfItems>102</NoOfItems>
     <Credit>72</Credit>
     <Timestamp>2007-01-16 16:29:00.978</Timestamp>
   </Order>
  ```

> **Note:** In the example, the local name of the top-level element should be same as the Data source name `<Order>`.
>
> Also, the indicators such as `Credit` are sent as child elements with the same name.

Message Element Defaults

- If `TargetName` and `TargetType` are part of the message, they should match the target name and type for the associated target (for that data source).
- If `TargetName` is not part of the message, it defaults to the target to which the data source was associated.
- If `TargetType` is not part of the message, it defaults to the target type of the target.
- If `Timestamp` is not included in the message, it defaults to the current timestamp.
- If `Category` is not included in the message, it defaults to the category `GenericExternalAlertMetric`.
- If `MetricName` is not included in the message, it defaults to the Alert metric.

- ProducerID is optional for the categories GenericExternalAlertMetric and Metric.

  However, producer ID is needed for user-defined metrics. In this case, ProducerID should be same as the metric author.

3.3.9.3 Inbound Alert Schema

External systems can send their own alerts/events to Enterprise Manager for display in the Enterprise Manager pages and be computed as part of SLA.

This schema is available in the following location:

$ORACLE_HOME/sysman/bam/InboundEvents.xsd

The schema of the incoming Alert message is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203/Inbo
undEvents/"
xmlns:de="http://xmlns.oracle.com/EnterpriseManager/DataExchange/10203/InboundEven
ts/" xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
      <!-- Define the Alert element -->
     <xs:element name="Alert" type="de:AlertType"/>
     <!-- Define the Alert Type -->
     <xs:complexType name="AlertType">
          <xs:all>
              <xs:element name="TargetType" type="xs:string" minOccurs="0"/>
              <xs:element name="TargetName" type="xs:string" minOccurs="0"/>
              <xs:element name="Category" type="xs:string" minOccurs="0"/>
              <xs:element name="MetricName" type="xs:string" minOccurs="0"/>
              <xs:element name="ProducerID" type="xs:string" minOccurs="0"/>
              <xs:element name="Severity" type="xs:string"/>
              <xs:element name="Message" type="xs:string" minOccurs="0"/>
              <xs:element name="Key1" type="xs:string" minOccurs="0"/>
              <xs:element name="Key2" type="xs:string" minOccurs="0"/>
              <xs:element name="Key3" type="xs:string" minOccurs="0"/>
              <xs:element name="Key4" type="xs:string" minOccurs="0"/>
              <xs:element name="Value" type="xs:string" minOccurs="0"/>
              <xs:element name="TimeStamp" type="xs:dateTime" minOccurs="0"/>
          </xs:all>
     </xs:complexType>
</xs:schema>
```

3.4 Integrating Enterprise Manager with OBAM

The following sections explain how to use the Data Exchange Connector to integrate OBAM with Enterprise Manager.

3.4.1 Supported Versions

The tested and certified versions of OBAM server are:

- Oracle BAM server 10gR2 (10.1.2.0.0) and 10gR2 patch sets
- Oracle BAM server 10gR3 (10.1.3.1.0) and 10gR3 patch sets
- Oracle BAM server 11gR1 (11.1.1.1.0) and 11gR1 patch sets

> **Note:** Oracle BAM Server 10gR2 and 10gR3 are OC4J-based. Oracle BAM Server 11gR1+ is WebLogic Web Server-based. Consequently, the setup steps differ considerably.
>
> The following sections provide basic steps and guidelines. Refer to the Oracle BAM Server documentation for specific information and details.

3.4.2 Setting up the Data Flow from Enterprise Manager to OBAM

For successful data flow from Enterprise Manager to OBAM, do the following:

1. Import required artifacts, explained in Importing OBAM Artifacts for an Outbound Session.
2. Update JNDI details, explained in Updating JNDI.
3. Run the link plans shown in Table 3–25. (This is only needed for OBAM 10g R3 and previous versions.)

3.4.2.1 Importing OBAM Artifacts for an Outbound Session

OBAM server is not packaged or installed as part of Enterprise Manager. It is assumed that an OBAM instance exists and is up and running. To read and persist the data from Enterprise Manager, certain artifacts should be existing and running. Import the artifacts from the pre-packaged scripts.

To import OBAM artifacts needed for the integration with OBAM server, as super user, run the following script:

- For Oracle BAM 11gR1 (11.1.1.1.0) or later versions:

  ```
  ICommand cmd=import file=emsys_all_11.xml
  ```

- For Oracle BAM 10gR3 or older versions:

  ```
  ICommand cmd=import file=emsys_all_10.xml
  ```

Both of these files are available at the following location:

```
$ORACLE_HOME/sysman/bam directory
```

The export script above creates the following OBAM artifacts:

- EM-BAM Data Objects
- EM-BAM EMS Definitions
- EM-BAM Enterprise Link Plans (only when emsys_all_10.xml is used). See Table 3–25.

EM-BAM Data Objects

Table 3–23 lists the data objects the Import command creates.

Table 3–23 EM-BAM Data Objects

Data Object	Description
/SYSMAN/EMSYSTargets	Contains target metadata information, such as target name and target type.
/SYSMAN/EMSYSMetrics	Contains metric metadata, such as metric name, metric column, and target type.

Table 3–23 (Cont.) EM-BAM Data Objects

Data Object	Description
/SYSMAN/EMSYSAlertsData	Contains the incoming system alerts received from Enterprise Manager. It contains information that includes alert message, alert severity, alert timestamp, and target information on which this alert has occurred.
/SYSMAN/EMSYSTargetSLA	Data object snapshot SLA values.
/SYSMAN/EMSYSTargetSLA Last24Hours	Data object to store the average SLA values for the last 24 hours.
/SYSMAN/EMSYSTargetSLA Last7Days	Data object to store the average SLA values for the last 7 days.
/SYSMAN/EMSYSTargetSLA Last31Days	Data object to store the average SLA values for the last 31 days.
/SYSMAN/EMSYSTargetSLAData	Contains the target SLA information received from Enterprise Manager.
/SYSMAN/EMSYSTargetStatus	Contains target status information, expressed as a percentage.
/SYSMAN/EMSYSTargetStatus Last24Hours	Contains target status information as the average value over 24 hours, expressed as a percentage.
/SYSMAN/EMSYSTargetStatus Last7Days	Contains target status information as the average value over 7 days, expressed as a percentage.
/SYSMAN/EMSYSTargetStatus Last31Days	Contains target status information as the average value over 31 days, expressed as a percentage.
/SYSMAN/EMSYSMetricsData	Data object for RAW metrics.
/SYSMAN/EMSYSMetricsData Last24Hours	Data object to store the average metrics data for the last 24 hours.
/SYSMAN/EMSYSMetricsData Last7Days	Data object to store the average metrics data for the last 7 days.
/SYSMAN/EMSYSMetricsData Last31Days	Data object to store the average metrics data for the last 31 days.
/SYSMAN/EMSYSSecurityFilter	Acts as the security filter for all other data objects. It contains the session name and users who can access the corresponding session data.

EM-BAM EMS Definitions

Table 3–24 lists the enterprise message sources the Import command creates.

Table 3–24 EM-BAM EMS Definitions

Data Definition	Description
EMSYSMetricsEMS	Contains the EMS definition for incoming metric metadata listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSMetricsTopic.
EMSYSTargetsEMS	Contains the EMS definition for incoming target metadata listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSTargetsTopic.
EMSYSSecurityFilterEMS	Contains the EMS definition for security filter data listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSSecurityFilterTopic.

Table 3–24 (Cont.) EM-BAM EMS Definitions

Data Definition	Description
EMSYSAlertsDataEMS	Contains the EMS definition for incoming alerts listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSAlertsDataTopic.
EMSYSMetricsDataEMS	Contains the EMS definition for incoming metrics listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSMetricsDataTopic.
EMSYSTargetStatusDataEMS	Contains the EMS definition for incoming target status messages metrics listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSTargetStatusTopic.
EMSYSTargetSLAEMS	Contains the EMS definition for incoming target SLA messages listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSTargetSLATopic.
EMSYSMetricsEMS-Queue	Contains the EMS definition for incoming metric metadata listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSMetricsQueue.
EMSYSTargetsEMS-Queue	Contains the EMS definition for incoming target metadata listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSTargetsQueue.
EMSYSSecurityFilterEMS-Queue	Contains the EMS definition for security filter data listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSSecurityFilterQueue.
EMSYSAlertsDataEMS-Queue	Contains the EMS definition for incoming alerts listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSAlertsDataQueue.
EMSYSMetricsDataEMS-Queue	Contains the EMS definition for incoming metrics listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSMetricsDataQueue.
EMSYSMetricsDataLast24Hours EMS-Queue	Contains the EMS definition for incoming metrics listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSMetricsDataLast24HoursQueue.
EMSYSMetricsDataLast7Days EMS-Queue	Contains the EMS definition for incoming metrics listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSMetricsDataLast7DaysQueue
EMSYSMetricsDataLast31Days EMS-Queue	Contains the EMS definition for incoming metrics listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSMetricsDataLast31DaysQueue
EMSYSMetricsDataEMS-Topic	Contains the EMS definition for incoming metrics listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSMetricsDataTopic.
EMSYSMetricsDataLast24Hours EMS-Topic	Contains the EMS definition for incoming metrics listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSMetricsDataLast24HoursTopic.
EMSYSMetricsDataLast7Days EMS-Topic	Contains the EMS definition for incoming metrics listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSMetricsDataLast7DaysTopic.
EMSYSMetricsDataLast31Days EMS-Topic	Contains the EMS definition for incoming metrics listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSMetricsDataLast31DaysTopic.

Table 3–24 (Cont.) EM-BAM EMS Definitions

Data Definition	Description
EMSYSTargetSLAEMS-Queue	Contains the EMS definition for incoming metrics listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSTargetSLAQueue.
EMSYSTargetSLALast24Hours EMS-Queue	Contains the EMS definition for incoming metrics listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSTargetSLALast24HoursQueue.
EMSYSTargetSLALast7Days EMS-Queue	Contains the EMS definition for incoming metrics listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSTargetSLALast7DaysQueue.
EMSYSTargetSLALast31Days EMS-Queue	Contains the EMS definition for incoming metrics listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSTargetSLALast31DaysQueue.
EMSYSTargetSLAEMS-Topic	Contains the EMS definition for incoming metrics listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSTargetSLATopic.
EMSYSTargetSLALast24Hours EMS-Topic	Contains the EMS definition for incoming metrics listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSTargetSLALast24HoursTopic.
EMSYSTargetSLALast7Days EMS-Topic	Contains the EMS definition for incoming metrics listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSTargetSLALast7DaysTopic.
EMSYSTargetSLALast31Days EMS-Topic	Contains the EMS definition for incoming metrics listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSTargetSLALast31DaysTopic.
EMSYSTargetStatusEMS-Queue	Contains the EMS definition for incoming metrics listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSTargetStatusQueue.
EMSYSTargetStatusLast24Hours EMS-Queue	Contains the EMS definition for incoming metrics listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSTargetStatusLast24HoursQueue.
EMSYSTargetStatusLast7Days EMS-Queue	Contains the EMS definition for incoming metrics listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSTargetStatusLast7DaysQueue.
EMSYSTargetStatusLast31Days EMS-Queue	Contains the EMS definition for incoming metrics listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSTargetStatusLast31DaysQueue.
EMSYSTargetStatusEMS-Topic	Contains the EMS definition for incoming metrics listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSTargetStatusTopic.
EMSYSTargetStatusLast24Hours EMS-Topic	Contains the EMS definition for incoming metrics listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSTargetStatusLast24HoursTopic.
EMSYSTargetStatusLast7Days EMS-Topic	Contains the EMS definition for incoming metrics listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSTargetStatusLast7DaysTopic.
EMSYSTargetStatusLast31Days EMS-Topic	Contains the EMS definition for incoming metrics listening on jms/EMSYSTopicConnectionFactory and jms/EMSYSTargetStatusLast31DaysTopic.
EMSYSTargetStatusDataEMS-Queue	Contains the EMS definition for incoming target SLA messages listening on jms/EMSYSQueueConnectionFactory and jms/EMSYSTargetSLAQueue.

3.4.2.2 Updating JNDI

You should update all EMSes described in Table 3–24 to reflect the correct JNDI initial context factory and provider URLs of your JMS servers. To do this:

1. In the OBAM Console, click **Architect.**

   The BAM Architect screen appears.

2. Select **Enterprise Message Sources** in the top left drop-down list.

   The left pane displays the six Enterprise Message Sources.

3. Click the **Message Source** links and do the following:

   a. In the right pane, click **Edit** and modify the JNDI Service Provider URL details from `t3://localhost` to the JNDI Service Provider URL of your Data Hub.

   b. Click **Save**.

4. Repeat this for all Message Source objects.

5. Make the following updates depending on the release:

   - For 11gR1 or new versions of OBAM:

     Enable Global trust or Cross domain trust between the Data hub WebLogic server and the OBAM WebLogic server. Refer to WebLogic documentation for details.

   - For 10gR3 or older versions of OBAM:

     Update `java.naming.security.principal` to the JMS server password in `jndi.properties` in the *`BAM_HOME`*`\BAM\j2re1.4.1_01\lib` directory.

6. Restart all OBAM services.

EM-BAM Enterprise Link Plans

Besides the BAM data objects (Table 3–23) and EMS definitions (Table 3–24), the link plans shown in Table 3–25 are also created when you use emsys_all_10.xml to create the artifacts. These are only needed for Oracle BAM Server 10gR3 or older versions. The plans shown in Table 3–25 are created based on the Import command.

Table 3–25 EM-BAM Link Plans

Plan	Description
EMSYSMetricsPlan	Contains the definition to receive, transform, and persist incoming metric metadata messages. This should be running before creating and setting up an outbound session in Enterprise Manager.
EMSYSTargetsPlan	Contains the definition to receive, transform, and persist incoming target metadata messages. This should be running before creating and setting up an outbound session in Enterprise Manager.
EMSYSSecurityFilterPlan	Contains the definition to receive, transform, and persist incoming security filter messages. This should be running before creating and setting up an outbound session in Enterprise Manager.

Table 3–25 (Cont.) EM-BAM Link Plans

Plan	Description
EMSYSAlertsDataPlan	Contains the definition to receive, transform, and persist incoming alert messages. This should be running when the outbound session with at least one selected alert is running in Enterprise Manager.
EMSYSMetricsDataPlan	Contains definition to receive, transform, and persist incoming metric messages. This should be running when the outbound session with at least one selected metric is running in Enterprise Manager.
EMSYSTargetStatusDataPlan	Contains definition to receive, transform, and persist incoming target status messages. This should be running when the outbound session with at least one availability selected metric is running in Enterprise Manager.
EMSYSTargetSLADataPlan	Contains definition to receive, transform, and persist incoming target SLA messages. This should be running when the outbound session with at least one SLA selected metric is running in Enterprise Manager.
EMSYSAlertsDataRollup	Contains the definition to move the data in EMSYSAlertsData that is more than 24 hours old to `EMSYSAlertsData.Archive`. You can run this based on demand.
EMSYSMetricsDataRollup	Contains the definition to move the data in EMSYSMetricsData that is more than 24 hours old to EMSYSMetricsData.Archive. You can run this based on demand.
EMSYSTargetStatusDataRollup	Contains the definition to move the data in EMSYSTargetStatusData that is more than 24 hours old to EMSYSTargetStatusData.Archive. You can run this based on demand.

3.4.3 Setting up the Data Flow from OBAM to Enterprise Manager

Unlike the Enterprise Manager to OBAM server data transfer, where pre-defined OBAM artifacts, such as Data Objects, EMS, and Plans (when needed) are defined and shipped along with Enterprise Manager, no such artifacts are defined or shipped for the data transfer from OBAM to Enterprise Manager. This is because the Data Object or EMS are unknown.

Consequently, for inbound data transfer from OBAM 11g or later versions, the integrator needs to use Oracle Data Integrator to read the data from the data object and put it on an outbound JMS topic or queue. For OBAM 10g or previous versions, the integrator can use outbound Plans to read data from data objects and put it on an outbound JMS topic or queue. Refer to ODI and OBAM documentation for specific information.

3.4.4 End-to-End Flow

After you configure the OBAM server with necessary artifacts and the JMS server with JMS topic or queue names, ensure the following for successful data flow from Enterprise Manager to OBAM:

- OBAM server and Enterprise Manager are up and running.
- JMS server, configured with required JMS topics or queues, is up and running.
- All enterprise plans (only for OBAM 10g or older versions) described in Table 3–25 are running.

  You can start plans manually or schedule them from Design Studio or through an alert.

After ensuring the specifications, proceed with the following:

- Create a data exchange hub and outbound data exchange session instances in Enterprise Manager.

  Always use a normalized message format for sessions that are for the consumption of the OBAM server. OBAM server plans are catered to understand only normalized messages.
- Schedule the outbound data exchange.

  Enterprise Manager sends data as scheduled in the outbound Data Exchange session.

3.5 Using an OC4J as a Data Exchange Hub

To use an OC4J as a data exchange hub instead of WebLogic Server, perform the following steps:

1. Copy the client libraries and restart Cloud Control:

   - For a 10.1.3.x OC4J, copy the 10.1.3.x `oc4jclient.jar` file to:

     ```
     $ORACLE_HOME/middleware/oms/sysman/archives/emgc/deployments/EMGC_DOMAIN/
     emgc.ear/APP-INF/lib
     ```

   - For a 10.1.2.x OC4J, copy the 10.1.2.x `oc4j.jar` file to:

     ```
     $ORACLE_HOME//middleware/oms/sysman/archives/emgc/deployments/EMGC_DOMAIN/
     emgc.ear/APP-INF/lib
     ```

2. Restart Cloud Control after copying the `.jar` file.
3. Configure the OC4J with the JMS destinations. You can do this with Application Server Control or by manually updating the `jms.xml` file as follows:

   The following example shows a `jms.xml` section for an OC4J:

   ```
       <topic-connection-factory/>
           location="jms/EMSYSTopicConnectionFactory"

       <topic>
           name="EMSYSAlertsDataTopic"
           location="jms/EMSYSAlertsDataTopic"
           <description>Topic for alerts data</description>
       </topic>

       <topic>
           name="EMSYSMetricsDataTopic"
           location="jms/EMSYSMetricsDataTopic"
           <description>Topic for metrics data</description>
       </topic>

       <topic>
   ```

```
    name="EMSYSMetricsTopic"
    location="jms/EMSYSMetricsTopic"
    <description>Topic for metrics metadata</description>
</topic>

<topic>
    name="EMSYSSecurityFilterTopic"
    location="jms/EMSYSSecurityFilterTopic"
    <description>Topic for security filter</description>
</topic>

<topic>
    name="EMSYSTargetStatusTopic"
    location="jms/EMSYSTargetStatusTopic"
    <description>Topic for target status</description>
</topic>

<topic>
    name="EMSYSTargetSLATopic"
    location="jms/EMSYSTargetSLATopic"
    <description>Topic for target SLA</description>
</topic>

<topic>
    name="EMSYSTargetsTopic"
    location="jms/EMSYSTargetsTopic"
    <description>Topic for targets metadata</description>
</topic>

<queue-connection-factory/>
    location="jms/EMSYSQueueConnectionFactory"

<queue>
    name="EMSYSAlertsDataQueue"
    location="jms/EMSYSAlertsDataQueue"
    <description>Queue for alerts data</description>
</queue>

<queue>
    name="EMSYSMetricsDataQueue"
    location="jms/EMSYSMetricsDataQueue"
    <description>Queue for metrics data</description>
</queue>

<queue>
    name="EMSYSMetricsQueue"
    location="jms/EMSYSMetricsQueue"
    <description>Queue for metrics metadata</description>
</queue>

<queue>
    name="EMSYSSecurityFilterQueue"
    location="jms/EMSYSSecurityFilterQueue"
    <description>Queue for security filter</description>
</queue>

<queue>
    name="EMSYSTargetStatusQueue"
    location="jms/EMSYSTargetStatusQueue"
    <description>Queue for target status</description>
```

```
</queue>

<queue>
    name="EMSYSTargetSLAQueue"
    location="jms/EMSYSTargetSLAQueue"
    <description>Queue for target SLA</description>
</queue>

<queue>
    name="EMSYSTargetsQueue"
    location="jms/EMSYSTargetsQueue"
    <description>Queue for targets metadata</description>
</queue>
```

3.6 Tips and Troubleshooting Information

The following sections provide various tips and troubleshooting information that might help in resolving various issues you encounter during the data exchange process.

3.6.1 Data Exchange Hub Connection Errors

A data exchange hub connection created in Cloud Control can error out due to authentication issues. These errors can appear in the following places:

- Cloud Control log files
- Cloud Control data exchange pages. For example:

  ```
  An error occurred while verifying the Data Exchange hub <hub_
  name>:
  You are not authorized to access the Data Exchange hub. The
  <session_name> session was not created successfully.
  ```

- Data exchange hub logs, such as an OC4J container error from OC4J logs. For example:

  ```
  2008-01-29 17:37:28.259 NOTIFICATION J2EE RMI-00004 Invalid
  username or password for default (oc4jadmin). Authentication
  failed.
  08/02/28 17:37:28 INFO: RMIProto  .readConnectionHeader Local
  ORMI version = 1. 3 different from remote version 1.1 will
  use 1.1
  2008-01-29 17:37:28.290 ERROR [RealmLoginModule]
  authentication failed
  ```

Follow these steps to resolve the connection problem:

1. Make sure the username/password combination is correct for the data exchange hub. You can check this with client programs, such as JDeveloper or a JMS client.
2. Recreate the Data Exchange hub connection entry in Cloud Control as follows:
   a. Log in to Cloud Control as a super user/administrator.
   b. From Enterprise Manager Cloud Control, click **Setup.**

      The setup links appear in the left margin.
   c. Click **Data Exchange**.

      The Data Exchange page appears.

d. In the Data Exchange Hub tab, select the hub connection and click **Delete**.

e. Create a new Data Exchange hub. See Creating a Data Exchange Hub.

3.6.2 Notification Methods and Rules

> **Important:** The verifications suggested in this section are meant for troubleshooting purposes and not for modification. Updating notification methods or rules can result in undesirable consequences.

Notification Method

For each data hub used for an outbound session, a new notification method is created. The name of the method is `*hub_name*NotifDevice`, where *`hub_name`* is the name of the data hub for which the method is created.

Notification Rules

For each outbound session (with Alerts selected), a notification rule is created.

The name of the rule is `*session_name*NotifRule`, where *`session_name`* is the name of the outbound session for which the rule is created.

To verify that the rule is successfully created:

1. From Enterprise Manager Cloud Control, click the **Preferences** link.
2. In the left pane under Notification, click **Rules.**

   The Notification Rules page appears.
3. Click on the corresponding Rule and make sure the selected targets and metrics are correct.

3.6.3 Data Flow Tips

- Ensure that the following JNDI details are correctly entered for the Data Hub you use:
  - JNDI URL
  - Username
  - Password
- For an inbound session setup, do not provide JNDI credentials for JMS in the data source definitions.

  If the JMS topic or queue is secured by providing authentication details, provide the username and password. If not, leave the fields blank.
- Ensure that the JMS server is up and running and configured with the required JMS topic and queue names.
- Ensure that either an inbound or outbound session is scheduled and is running.
- For an inbound session data source, to ensure that the topic details are correctly entered, click **Test Connection**.
- Using the same topic or queue name for two active inbound sessions could result in corrupted data.

- Frequency for an inbound session should either be synchronized with or relative to the frequency at which the external system sends data.

  For example, setting the inbound session frequency to 2 minutes is ineffective if the external system sends data only once in 10 minutes.

- For an outbound session, ensure that Receiver or EL Plans (in the case of OBAM) are running.
- To improve efficiency, outbound session schedule frequency should be relative or synchronized with the frequencies at which Enterprise Manager collects the metrics defined in the session.

  For example, in Enterprise Manager, if the collection frequency for metrics defined in the outbound session is 5 minutes, setting a lesser outbound frequency (one minute, for instance) is ineffective, as the possibility of new data is remote. In such cases, setting the outbound frequency to 5 or 10 minutes would be effective.

  Note that only new metric values (if any) are forwarded.

- Do not schedule two or more outbound sessions with different message formats at the same time.
- Unless a lower frequency level is absolutely required, always set higher frequency intervals. This helps to reduce the Enterprise Manager/JMS server load.

3.6.4 Log Files

Always check the log if data could not be received or if the status is `Failure`. To check logs:

1. From Enterprise Manager Cloud Control, click the **Jobs** tab.
2. Click **Advanced Search.**
3. In Target Type, select **Targetless** from the drop-down list.
4. In Status, select **All** from the drop-down list.
5. Click **Go.**
6. Click the Job you want to verify.
   - For an inbound session, the name of the job is `<Session Name>ISJOB`.
   - For an outbound session, the name of the job is `<Session Name>OSJOB`.
7. Click the Status value link; for instance, `Succeeded` or `Failed`.
8. Click **Step.**
   - For inbound sessions, the step name is `receiveMetricDataViaJms`.
   - For outbound sessions, the step name is `sendMetricDataViaJms`.
9. Note the Step ID Value and search logs (typically located in the directory `$ORACLE_HOME/sysman/log`) based on the ID in the log directory using the following command:

   ```
   "grep -i "JobWorker stepID" *.trc
   ```

   Note: The default system log directory is `$ORACLE_HOME/sysman/log`

10. Check for the following:

    - Exceptions or errors
    - `emoms.log` (in the same directory) for any other exception for the same timestamp

11. Change the log level in `emomslogging.properties` to `DEBUG` and restart Enterprise Manager for more detailed debugging information.

3.6.5 End-to-End Flow Sample Demonstrations

For the convenience of integrators, Oracle provides sample demonstrations detailing the end-to-end data flow. To access the demonstrations, go to the following directory:

```
$ORACLE_HOME/sysman/bam
```

Instructions for an outbound session sample are provided in the file `outboundsession_sample_readme.txt`. You can create the report required for the demonstration by importing the file `outboundsession_report.xml` as detailed in the readme file.

Instructions for an inbound session sample are provided in the file `inboundsession_sample_readme.txt`. You can create the artifacts required for the demonstration by importing the file `inboundsession.xml` as detailed in the readme file.

3.7 Suggested Reading

The following list provides online resources that can help you effectively use all associated technologies involved in the data exchange process.

- Oracle Business Activity Monitoring: http://www.oracle.com/technology/products/integration/bam/index.html
- Oracle Data Integrator:

  http://www.oracle.com/technology/products/oracle-data-integrator/index.html
- Java Message Service:

  http://java.sun.com/products/jms/
- Oracle Containers for J2EE (OC4J):

  http://www.oracle.com/technology/tech/java/oc4j/index.html
- Oracle Enterprise Service Bus (ESB):

  http://www.oracle.com/technology/products/integration/esb/index.html

- Enterprise Manager Metrics:

  Oracle Enterprise Manager Framework, Host, and Third-Party Metric Reference Manual available at the following URL:

  `http://www.oracle.com/technology/documentation/oem.html`

4 Reference Tables

This chapter provides tabular reference information for connectors. The following sections provide reference tables for the following categories:

- Request Attributes
- Response File Properties for the Windows Platform
- Queryable Properties
- Complex Response Properties
- Status Codes

This chapter also provides information about the response file properties that the `Create RAC` and `Add Node` jobs generate for the Windows platform.

4.1 Request Attributes

The tables in this section provide query paths, descriptions, and data types for the following property types:

- setModel Request
- Request Header
- Create RAC
- Add Node
- Delete Node

Table 4–1 provides path types, descriptions, and data types for setModel request elements.

Table 4–1 setModel Request Elements

Path Type	Description	Data Type
<EMModel xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance xmlns=**"http://xmlns.oracle.com/sysman/connector/base/msi"** >	EMModel element.	Complex Type
<RequestHeader/>	Request header. See Table 4–2.	Complex Type
<Credential>	Credential for Enterprise Manager.	Complex Type
<Name>sysman</Name>	User name.	String

Table 4–1 (Cont.) setModel Request Elements

Path Type	Description	Data Type
<Password>welcome1 </Password>	Password.	String
</Credential>	End of Credential.	Complex Type
<AggregateTarget>	EMModel should contain two aggregate targets: one of type cluster, and one of type rac_database. This aggregate target is of type cluster. It contains information about the cluster.	Complex Type
<Name>CRS30</Name>	Name of the cluster.	String
<Type>cluster</Type>	Aggregate target type.	String (enumeration: "cluster" "rac_database"
<Target>	Number of targets in the cluster aggregate target should be the same as the number of hosts in the cluster. Each target element contains information about a host in the cluster.	Complex Type
<Name>bjx30</Name>	Name of the host.	String
<Type>host</Type>	Target type. In the cluster aggregate target, it should be set to "host."	String
<Host>bjx30</Host>	Name of the host.	String
<Credential>	Credential of the host.	Complex Type
<Name>oracle</Name>	User name.	String
<Password>welcome1 </Password>	Password.	String
</Credential>	End of Credential.	Complex Type
<Property/>	Property of the target. See the corresponding table in this section for the properties of the "host" target type.	Complex Type
</Target>	End of Target.	Complex Type
<Property/>	Property of the cluster aggregate target. See the corresponding table in this section for the properties of the "cluster" aggregate target type.	Complex Type
</AggregateTarget>	End of AggregateTarget.	Complex Type
<AggregateTarget>	EMModel should contain two aggregate targets: one of type cluster and one of type rac_database. This aggregate target is of type rac_database. It contains information about the RAC.	Complex Type
<Name>RAC30</Name>	Name of the RAC database.	String (length <=8)
<Type>rac_database</Type>	Aggregate target type.	String (enumeration: "cluster" or "rac_database")
<Storage>	Storage element contains storage information for the RAC. This element can be omitted for the Add Node job request if the storage type is not ASM.	Complex Type
<Type>ASM</Type>	Type of storage.	String (enumeration: "CFS" or "ASM." "RAW" is not supported.)
<Property/>	Properties for storage. See the corresponding table in this section for the storage properties.	Complex Type

Table 4–1 (Cont.) setModel Request Elements

Path Type	Description	Data Type
</Storage>	End of Storage.	Complex Type
<Target>	The number of targets in the `rac_database` aggregate target should be the same as the number of database instances in the cluster database. Each target element contains information about a database instance in the RAC database.	Complex Type
<Name>RAC30_RAC301</Name>	Name of the target, which should be in the format of `<rac_name>_<rac_instance_name>`.	String
<Type>oracle_database</Type>	Target type. In the `rac_database` aggregate target, it should be set to "`oracle_database`."	String
<Host>bjx30</Host>	Name of the host.	String
<Credential>	Credential of the host.	Complex Type
<Name>oracle</Name>	User name.	String
<Password>welcome1 </Password>	Password.	String
</Credential>	End of Credential.	Complex Type
<Property/>	Property of the target. See the corresponding table in this section for the properties of the "`oracle_database`" target type.	Complex Type
</Target>	End of Target.	Complex Type
<Property/>	Property of the `rac_database` aggregate target. See the corresponding table in this section for the properties of the "`rac_database`" aggregate target type.	Complex Type
</AggregateTarget>	End of AggregateTarget.	Complex Type
</EMModel>	End of EMModel.	Complex Type

Table 4–2 provides path types, descriptions, and data types for request header elements.

Table 4–2 Request Header Elements

Path Type	Description	Data Type
<RequestID/>	Uniquely identifies the request. This is mainly used by the client. Enterprise Manager currently does not track this ID.	String
<Source/>	Request source, which is the request operating system.	String
<Destination/>	Destination should be Enterprise Manager in this release.	String
<RequestProperty> <Type>Singleton</Type> <Property> <Name>Platform</Name> <Value>Linux</Value> </Property> </RequestProperty>	Platform is an optional property. Specify either Linux or Windows. If you do not specify a platform, the default is Linux. The platform is only relevant in provisioning use cases.	

Table 4–3 provides target types, properties, descriptions, and data types for Create RAC.

Table 4–3 Create RAC Properties

Target Type	Property	Description	Data Type
<Target> <Type>host</Type> </Target>	CRS_HOME	Oracle Clusterware home directory. This property must be the same for all hosts in the cluster.	String
	ORACLE_HOME_NAME	Oracle Clusterware home name. This is an optional property. The default value is the cluster aggregate target name.	String
	publicNode	Public node name.	String
	privateNode	Private node name.	String
	vipNode	Virtual node name.	String
<Target> <Type>oracle_database</Type> </Target>	ORACLE_HOME	Database home directory. This property must be the same for all database instances in the RAC database.	String
	ORACLE_HOME_NAME	Database home name. This property is optional. The default value is the `rac_database` aggregate target name.	String
	db_username	Database user for setting monitoring credentials. It should always be SYS in this release.	String
	db_password	Password of the database user. It is the password for SYS and SYSTEM in this release.	String
	oms_username	Oracle Management Services host operating system user name.	String
	oms_password	Oracle Management Services host operating system password.	String
<Storage> <Type>ASM</Type> </Storage>	diskGroupName	ASM disk group name.	String
	diskList	Disk device list. Use a comma (,) as a separator.	String (no space allowed)
	diskString	Search paths for ASM disks.	String (no space allowed)
	redundancy	Redundancy level.	String (enumeration: NORMAL, HIGH, or EXTERNAL)
	asmPassword	ASM SYSDBA password.	String

Table 4–3 (Cont.) Create RAC Properties

Target Type	Property	Description	Data Type
<Storage> <Type>CFS</Type> </Storage>	datafileDestination	Data file directory.	String
<AggregateTarget> <Type>cluster</Type> </AggregateTarget>	softwareImageName	Name of the software library image for the CRS home. This property is optional. The default value is the latest active software library image of type "Oracle Clusterware Clone."	String
	s_ocrpartitionlocation	OCR location. Use the comma (,) as a separator. This property is only for the Linux platform.	String (no space allowed)
	s_votingdisklocation	Voting disk location. Use the comma (,) as a separator. This property is only for the Linux platform.	String (no space allowed)
	RESPONSEFILE_VERSION	Response file version. This property is optional. The default value is 2.2.1.0.0. This property is only for the Windows platform.	String
	sl_OHPartitionsAndSpace_valueFromDlg	This property specifies the split-up of disk partitions for OCR/Vdisk locations. This property is only for the Windows platform. See Section 4.2, "Response File Properties for the Windows Platform" for more information.	String
	ret_PrivIntrList	This property specifies the interconnects to use. This property is only for the Windows platform. See Section 4.2, "Response File Properties for the Windows Platform" for more information.	String
<AggregateTarget> <Type>rac_database</Type> </AggregateTarget>	templateName	Database template name. This property is optional. The default value is `General_Purpose.dbc`	String (file name without path)
	gdbName	Global database name. This property is optional. The default value is the `rac_database` aggregate target name.	String (length <=8)

Table 4–3 (Cont.) Create RAC Properties

Target Type	Property	Description	Data Type
	sid	Database instance name. This should be the same as `gdbName` in this release. This property is optional. The default value is the `rac_database` aggregate target name.	String (length <=8)
	characterSet	Character set. See the Database Globalization Support guide for details. This property is optional. The default value is UTF8.	String
	nationalCharacterSet	National character set. See the Database Globalization Support guide for details. This property is optional. The default value is UTF8.	String

	initParams	Raw strings for additional input. For example:
		`nls_territory=japan, nls_language=japanese`
		This property is optional. The default value is:
		`nls_lang=american,nls_territory=american`
			String (no space allowed)

	softwareImageName	Name of the software library image for the database home. This property is optional. The default value is the latest active software library image of type "Oracle Database Clone."	String

Table 4–4 provides target types, properties, descriptions, and data types for Add Node for a RAC aggregate target.

Table 4–4 Add Node Properties (Storage and Aggregate Target)

Target Type	Property	Description	Data Type
<Storage> <Type>ASM</Type> </Storage>	asmPassword	ASM SYSDBA password.	String
<AggregateTarget> <Type>cluster</Type> </AggregateTarget>	s_ocrpartitionlocation	OCR location. Use the comma (,) as a separator. This property is only for the Linux platform. Set this value to be the same as the Create RAC job.	String (no space allowed)
	s_votingdisklocation	Voting disk location. Use the comma (,) as a separator. This property is only for the Linux platform. Set this value to be the same as the Create RAC job.	String (no space allowed)

Table 4–4 (Cont.) Add Node Properties (Storage and Aggregate Target)

Target Type	Property	Description	Data Type
	RESPONSEFILE_VERSION	Response file version. This property is optional. The default value is 2.2.1.0.0. This property is only for the Windows platform. Set this value to be the same as the Create RAC job.	String
	sl_OHPartitionsAndSpace_valueFromDlg	This property specifies the split-up of disk partitions for OCR/Vdisk locations. This property is only for the Windows platform. Set this value to be the same as the Create RAC job.	String
	ret_PrivIntrList	This property specifies the interconnects to use. This property is only for the Windows platform. Set this value to be the same as the Create RAC job.	String

Table 4–5 provides target types, properties, descriptions, and data types for Add Node for a new node.

Table 4–5 Add Node Properties (New Node)

Target Type	Property	Description	Data Type
<Target> <Type>host</Type> </Target>	CRS_HOME	Oracle Clusterware home directory. This property must be the same for all hosts in the cluster.	String
	ORACLE_HOME_NAME	Oracle Clusterware home name. This property is optional. The default value is cluster aggregate target name.	String
	publicNode	Public node name.	String
	privateNode	Private node name.	String
	vipNode	Virtual node name.	String
<Target> <Type>oracle_database</Type> </Target>	ORACLE_HOME	Database home directory. This property must be the same for all database instances in the RAC database.	String
	ORACLE_HOME_NAME	Database home name. This property is optional. The default value is `rac_database` aggregate target name.	String
	db_username	Database user name that has a SYSDBA role.	String

Table 4–5 (Cont.) Add Node Properties (New Node)

Target Type	Property	Description	Data Type
	db_password	Password of the user above.	String
	oms_username	Oracle Management Services host operating system user name.	String
	oms_password	Oracle Management Services host operating system password.	String

Table 4–6 provides target types, properties, descriptions, and data types for Add Node for any existing node.

Table 4–6 Add Node Properties (Any Existing Node)

Target Type	Property	Description	Data Type
<Target> <Type>host</Type> </Target>	publicNode	Public node name.	String
	CRS_HOME	Oracle Clusterware home directory. This property must be the same for all hosts in the cluster.	String
<Target> <Type>oracle_database</Type> </Target>	ORACLE_HOME	Database home directories. This property must be the same for all database instances in the RAC database.	String

Table 4–7 provides target types, properties, descriptions, and data types for Delete Node for remaining nodes.

Table 4–7 Delete Node Properties (Remaining Nodes)

Target Type	Property	Description	Data Type
<Target> <Type>host</Type> </Target>	CRS_HOME (or ORACLE_HOME)	Oracle Clusterware home directory.	String
<Target> <Type>oracle_database</Type> </Target>	ORACLE_HOME	Database home directory.	String
	db_username	Database user name that has a SYSDBA role.	String
	db_password	Password of the user above.	String

Table 4–7 (Cont.) Delete Node Properties (Remaining Nodes)

Target Type	Property	Description	Data Type
	oms_username	Oracle Management Services host operating system user name.	String
	oms_password	Oracle Management Services host operating system password.	String
<AggregateTarget> <Type>rac_database</Type> </AggregateTarget>	oms_delete_all_targets	Removes all targets on the instance host, including the host and agent. This property is optional. The default value is False.	String (enumeration: True or False)

4.2 Response File Properties for the Windows Platform

The following properties are required to generate the response file for the `Create RAC` and `Add Node` jobs on the Windows platform:

- `sl_OHPartitionsAndSpace_valueFromDlg`
- `ret_PrivIntrList`

The following sections describe each response file property.

4.2.1 sl_OHPartitionsAndSpace_valueFromDlg Property

This property specifies the splitting up of disk partitions for OCR/Vdisk locations. It consists of the following six values for each location:

- Disk no.
- Partition no.
- Partition Size (MB)
- Format Type
  - 0: None/RAW
  - 1: CFS for data
  - 2: CFS for software
- Drive Letter
  - N/A: RAW
  - "Available" drive letter: CFS
- Usage Type
  - 0: Data/software use ONLY
  - 1: OCR primary ONLY
  - 2: Voting disk ONLY
  - 3: OCR primary and voting disk on the same CFS partition
  - 4: OCR mirror only
  - 5: OCR mirror and voting disk on the same CFS partition

Example 1

Given the following scenario:

- OCR and the Voting Disk are on Partition-2 of Disk-1 (Partition-2 has size 10002 MB).
- The partition is CFS-formatted.
- Both OCR and the Voting Disk reside on the same partition.
- The drive letter for the partition is `G:`.
- There is only data storage and no software storage.

You would specify `sl_OHPartitionsAndSpace_valueFromDlg` as follows:

```
<Property>
    <Name>sl_OHPartitionsAndSpace_valueFromDlg</Name>
    <Value>{"1","2","10002","1","G:","3"}</Value>
</Property>
```

Example 2

Given the following scenario:

- OCR and the Voting Disk reside on different partitions.
- OCR is on Partition-1 of Disk-3, which has a size of 486 MB and is RAW-formatted.
- The Voting Disk is on Partition-1 of Disk-4, which has a size of 486 MB and is RAW-formatted.

You would specify `sl_OHPartitionsAndSpace_valueFromDlg` as follows:

```
<Property>
    <Name>sl_OHPartitionsAndSpace_valueFromDlg</Name>
    <Value>{"3","1","486","0","N/A","1","4","1","486","0","N/A","2"}</Value>
</Property>
```

4.2.2 ret_PrivIntrList Property

This property specifies the interconnects to use. You should specify entries in `ret_PrivIntrList` as a comma-separated list of interfaces. Each entry should be a colon-separated string with three fields. You should specify the fields as follows:

- The first field should be the interface name.
- The second field should be the subnet IP of the interface.
- The third field should indicate how Oracle Clusterware should use the interface: as a public interface, private interface, or whether it should not be used at all by the clusterware. This field should be specified as a number — 1, 2, or 3. These numbers represent the following values:
  - 1: Public
  - 2: Private
  - 3: Do not use

Example

Given the following scenario:

- One "Local Area Connection" public interconnect is to be used.

- One "Local Area Connection2" private interconnect is to be used.

You would specify `ret_PrivIntrList` as follows:

```
<Property>
    <Name>ret_PrivIntrList</Name>
    <Value>{"Local Area Connection:123.45.67.0:1","Local Area Connection
2:123.45.89.0:2"}</Value>
</Property>
```

4.3 Queryable Properties

The tables in this section provide property names, descriptions, and data types for the following types of queryable properties:

- General Target
- Oracle Database
- Oracle Listener
- Host Target
- Cluster
- Cluster Database
- Oracle Enterprise Manager Agent
- Oracle Enterprise Manager Repository Target
- Job

Table 4–8 provides property names, descriptions, and data types for general target queryable properties.

Table 4–8 General Target Properties

Query Path	Description	Data Type
Property(Name:status)	Integer status of the Enterprise Manager target instance. (See Table 4–20.)	Integer
Property(Name:monitoring agent)	Enterprise Manager target instance name (of type `oracle_emd`) of the Agent monitoring the Enterprise Manager target instance.	String
Property(Name:homepage)	Enterprise Manager Console home page URI (the path portion of the URL, as in `/em/console?...`) of the Enterprise Manager target instance.	URL
Property(Name:version)	Version of the Enterprise Manager target instance.	String
Property(Name:oracle home)	Oracle home of the Enterprise Manager target instance. The form of the directory path (path separator) is not further specified here.	String
Property(Name:critical alerts)	Number of critical alerts against the Enterprise Manager target instance.	Integer
Property(Name:warning alerts)	Number of warning alerts against the Enterprise Manager target instance.	Integer
Property(Name:critical policy violations)	Number of critical policy violations against the Enterprise Manager target instance.	Integer
Property(Name:warning policy violations)	Number of warning policy violations against the Enterprise Manager target instance.	Integer

Table 4–8 (Cont.) General Target Properties

Query Path	Description	Data Type
Property(Name:compliance score)	Compliance score as a real number between 0 and 1 (inclusive) of the Enterprise Manager target instance.	Number
Property(Name:last load time)	Last load time of the data for the target instance as the number of milliseconds since January 1, 1970, 00:00:00 GMT.	Number
Host	Enterprise Manager target instance name (of type host) of the host related to the Enterprise Manager target instance.	String

Table 4–9 provides property names, descriptions, and data types for Oracle Database queryable properties. The target type for the Oracle Database is `oracle_database`.

Table 4–9 Oracle Database Properties

Query Path	Description	Data Type
Property(Name:instance name)	Instance name of the database instance.	String
Property(Name:listener)	Listener Enterprise Manager target instance name (of type `oracle_listener`) of the listener for the database instance.	String
Property(Name:is archiving)	The value is 1 if high availability archiving is on for the Oracle database instance. Otherwise, the value is 0.	Integer
Property(Name:is flashback logging)	The value is 1 if high availability flashback logging is on for the Oracle database instance. Otherwise, the value is 0.	Integer

Table 4–10 provides property names, descriptions, and data types for Oracle listener properties. The target type for the Oracle listener is `oracle_listener`.

Table 4–10 Oracle Listener Properties

Query Path	Description	Data Type
Property(Name:alias)	Alias of the Oracle Listener instance.	String
Property(Name:net address)	Net address of the Oracle Listener instance.	URI
Property(Name:listener.ora location)	File directory location of the `listener.ora` file of the Oracle Listener instance. The form of the directory path (path separator) is not further specified here.	String
Property(Name:start name)	Start time of the Oracle Listener instance. The form of this time stamp is not further specified here.	Time

Table 4–11 provides property names, descriptions, and data types for host target properties. The target type for the Host is host.

To get the targets within the domain of a cluster, first request the cluster hosts with the "Target" sub-element. Then get all the targets and filter the list by the hosts in the cluster hosts list.

Table 4–11 Host Target Properties

Query Path	Description	Data Type
Property(Name:cluster)	Enterprise Manager target instance name (of type cluster) of the cluster for this host instance.	String
Property(Name:cpu utilization)	CPU utilization as a real number between 0 and 1 (inclusive) of the host.	Number
Property(Name:memory utilization)	Memory utilization as a real number between 0 and 1 (inclusive) of the host.	Number
Property(Name:total io rate)	Total I/O per second.	Number

Table 4–12 provides property names, descriptions, and data types for cluster properties. The target type for Oracle Clusterware is cluster.

Table 4–12 Cluster Properties

Query Path	Description	Data Type
Property(Name:version)	Clusterware version. Note that this property definition just redefines the same property defined for the general target mappings.	String
Property(Name:cluster database)	Cluster databases (of type `rac_database`).	String
Target	Cluster hosts (of type host).	String

Table 4–13 provides property names, descriptions, and data types for cluster database properties. The target type for the Oracle cluster database is `rac_database`.

Table 4–13 Cluster Database Properties

Query Path	Description	Data Type
Property(Name:cluster)	Enterprise Manager target instance name (of type cluster) of the cluster for this database instance.	String
Property(Name:database name)	Database instance name.	String
Property(Name:is archiving)	Value is 1 if high availability archiving is on for the cluster database. Otherwise, the value is 0.	Integer
Target	Cluster database instance of type `oracle_database`.	String

Table 4–14 provides property names, descriptions, and data types for Oracle Enterprise Manager Agent properties. The target type for the Oracle Management Agent is `oracle_emd`.

Table 4–14 Oracle Enterprise Manager Agent Properties

Query Path	Description	Data Type
Property(Name:management service)	OMS that the Enterprise Manager Agent instance uploads to.	String

Table 4–15 provides property names, descriptions, and data types for Oracle Enterprise Manager Repository target properties. The target type for the Oracle Management Repository is `oracle_emrep`.

Table 4–15 Oracle Enterprise Manager Repository Target Properties

Query Path	Description	Data Type
Property(Name:agent count)	Number of Agents for this repository instance.	Integer
Property(Name:target count)	Number of targets for this repository instance.	Integer
Property(Name: administrator count)	Number of administrators for this repository instance.	Integer
Property(Name:session count)	Number of active Oracle management services repository sessions for this repository instance.	Integer
Property(Name:Integer)	Enterprise Manager database target instance(s) of the database(s) for this repository instance. This property is expanded into complex property elements in the response as described in Table 4–18. They are keyed by the "name" and "value" sub-properties.	String
Property(Name:tablespace)	Expands to the tablespaces used in the database for this repository instance.	String
Property(Name:oms)	OMSs for this Enterprise Manager repository. This property is expanded into complex property elements in the response as described in Table 4–17. They are keyed by the "name" sub-properties.	String

Table 4–16 provides property names, descriptions, and data types for job properties.

Table 4–16 Job Properties

Query Path	Description	Data Type
JobStatus	Integer status (see Table 4–20) of the most recent execution of the job.	Integer
Property(Name:output)	Last 1024 characters of the job output for the last step of the most recent job execution.	String

4.4 Complex Response Properties

The tables in this section provide property names, descriptions, and data types for the following complex properties returned in the response model to the query requests:

- Oracle Management Service (OMS)
- Database instance

Table 4–17 provides property names, descriptions, and data types for the Oracle Management Service. The type of the complex property is OMS.

Table 4–17 Oracle Management Service (OMS) Complex Property

Query Path	Description	Data Type
ComplexProperty(Type:oms). Property(Name:name)	OMS name.	String
ComplexProperty(Type:oms). Property(Name:status)	OMS service status (1 for up or 0 for down) for the OMS given by the peer "name" property.	Integer
ComplexProperty(Type:oms). Property(Name:last error)	Time of the last OMS error as the number of milliseconds since January 1, 1970, 00:00:00 GMT, for the OMS given by the peer "name" property.	Number

Table 4–17 (Cont.) Oracle Management Service (OMS) Complex Property

Query Path	Description	Data Type
ComplexProperty(Type:oms). Property(Name:files pending load)	Number of files pending loading into the OMS for the OMS given by the peer "name" property.	Integer
ComplexProperty(Type:oms). Property(Name:load directory)	Load directory of the OMS for the OMS given by the peer "name" property. The form of the directory path (path separator) is not specified further here.	String
ComplexProperty(Type:oms). Property(Name:oldest load file)	Oldest file to load (in minutes) of the OMS for the OMS given by the peer "name" property.	Number

Table 4–18 provides property names, descriptions, and data types for the database instance. The type of the complex property is OMS.

Table 4–18 Database Instance Complex Property

Response Path	Description	Data Type
ComplexProperty(Type:database). Property(Name:name)	Oracle database instance name.	String
ComplexProperty(Type:database). Property(Name:type)	Oracle database instance type (one of `oracle_database` or `rac_database`).	String

4.5 Status Codes

The following tables provide status codes and descriptions for the following status types:

- Enterprise Manager
- Jobs

Table 4–19 describes the status codes for Enterprise Manager. You can use online help for a detailed description of Enterprise Manager target statuses. Enter **Target Status** as the keywords to search in online help, then select the topic **About the Status Icons**.

Table 4–19 Enterprise Manager Status Codes

Status Code	Description
0	Target down
1	Target up
2	Metric error
3	Agent down
4	Unreachable
5	Blackout
6	Pending/unknown

Table 4–20 describes the status codes for jobs. You can use online help for a detailed description of Enterprise Manager job statuses. Enter **Job Status** as the keywords to search in online help, then select the topic **About Job Status**.

Table 4–20 Job Status Codes

Status Code	Description
1	SCHEDULED — The execution is in a scheduled state.
2	RUNNING — The execution is running.
3	INITIALIZATION ERROR — The execution encountered an error and the remote process did not run.
4	FAILED — The execution failed.
5	SUCCEEDED — The execution succeeded.
6	SUSPENDED BY USER — A user suspended the execution.
7	SUSPENDED ON AGENT UNREACHABLE — The execution was suspended because the Agent was unreachable.
8	STOPPED — A user stopped the execution.
9	SUSPENDED ON LOCK — The execution is waiting for a lock on a shared resource.
10	SUSPENDED ON EVENT — The execution is waiting for an event to occur (usually for an Agent to bounce).
11	SUSPENDED ON BLACKOUT — The execution is suspended on a blackout.
12	STOP PENDING — The execution is in Stop Pending status waiting for some running steps to finish.
13	SUSPEND PENDING — The execution is in Suspend Pending status waiting for some running steps to finish.
14	Inactive (internal state).
15	Queued (internal state).
16	Failed retried (internal state).
18	SKIPPED — The execution was skipped and could not run, because the previous run of the job required too much time, or the Agent was unavailable for too long a period of time.
20	REASSIGNED — The execution is suspended because the original owner of the job was deleted and the job is not assigned to a new owner. The new owner must explicitly resume the job from the console.
21	SUSPENDED ON MISSING CREDENTIALS — The execution is suspended because some of the credentials needed for the job are not set.

A

Ticketing Connector Samples

This appendix provides sample implementations for Remedy 7 Help Desk ticketing connectors.

Example A–1 connectorDeploy.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<ManagementConnector xmlns="http://xmlns.oracle.com/sysman/connector">
<Name>Remedy Service Desk Connector</Name>
<Version>11.2.0.1.0</Version>
<EMCompatibleVersion>11.2.0.1.0</EMCompatibleVersion>
<Description>Remedy Integration with Enterprise Manager</Description>
<Category>TicketingConnector</Category>
<SOAPHeaderAuthentication>
<Username required="true">
<VariableName>USERNAME</VariableName>
<DisplayName>Remedy Username</DisplayName>
</Username>
<Password>
<VariableName>PASSWORD</VariableName>
<DisplayName>Remedy Password</DisplayName>
</Password>
<AuthVariable>
<VariableName>AUTHENTICATION</VariableName>
<DisplayName>Authentication</DisplayName>
</AuthVariable>
<AuthVariable>
<VariableName>LOCALE</VariableName>
<DisplayName>Locale</DisplayName>
</AuthVariable>
<AuthVariable>
<VariableName>TIMEZONE</VariableName>
<DisplayName>Timezone</DisplayName>
</AuthVariable>
<SOAPHeader>
<![CDATA[
<urn:AuthenticationInfo xmlns:urn="urn:HelpDesk_Submit_Service">
<urn:userName>$USERNAME$</urn:userName>
<urn:password>$PASSWORD$</urn:password>
<urn:authentication>$AUTHENTICATION$</urn:authentication>
<urn:locale>$LOCALE$</urn:locale>
<urn:timeZone>$TIMEZONE$</urn:timeZone>
</urn:AuthenticationInfo>
     ]]>
</SOAPHeader>
</SOAPHeaderAuthentication>
```

```
<ConnectivityTestVariable>
<VariableName>TICKET_ID</VariableName>
<DisplayName>Ticket ID</DisplayName>
</ConnectivityTestVariable>
<Service>
<Method>createTicket</Method>
<WebServiceEndpoint>
<![CDATA[http://[midtier_
server]/arsys/services/ARService?server=[servername]&webService=HPD_
IncidentInterface_Create_WS]]>
</WebServiceEndpoint>
</Service>
<Service>
<Method>updateTicket</Method>
<WebServiceEndpoint>
<![CDATA[http://[midtier_
server]/arsys/services/ARService?server=[servername]&webService=HPD_
IncidentInterface_WS]]>
</WebServiceEndpoint>
</Service>
<Service>
<Method>getTicket</Method>
<WebServiceEndpoint>
<![CDATA[http://[midtier_
server]/arsys/services/ARService?server=[servername]&webService=HPD_
IncidentInterface_get_WS]]>
</WebServiceEndpoint>
</Service>
<ExternalURL>
<Pattern>
<![CDATA[http://$WEB_SERVER$/arsys/forms/$ARSERVER_NAME$/$FORM_
NAME$/?qual=%27Incident%20Number*%27=%22@Incident_Number@%22]]>
</Pattern>
<ConfigVariable required="true">
<VariableName>WEB_SERVER</VariableName>
<DisplayName>Web Server</DisplayName>
</ConfigVariable>
<ConfigVariable required="true">
<VariableName>FORM_NAME</VariableName>
<DisplayName>HelpDesk Case Form Name</DisplayName>
</ConfigVariable>
<ConfigVariable required="true">
<VariableName>ARSERVER_NAME</VariableName>
<DisplayName>ARServer Name</DisplayName>
</ConfigVariable>
</ExternalURL>
<TemplateRegistration>
<FileName>getTicket_request.xml</FileName>
<InternalName>getTicket</InternalName>
<TemplateName>Get Ticket</TemplateName>
<TemplateType>OutboundXML</TemplateType>
<Description>This is the request xml file for getTicket method</Description>
</TemplateRegistration>
<TemplateRegistration>
<FileName>getTicket_response.xsl</FileName>
<InternalName>getTicket</InternalName>
<TemplateName>Get Ticket</TemplateName>
<TemplateType>InboundXSL</TemplateType>
<Description>This is the response xsl file for getTicket method</Description>
</TemplateRegistration>
```

```
<TemplateRegistration>
<FileName>createTicket_response.xsl</FileName>
<InternalName>createTicket</InternalName>
<TemplateName>Create Ticket Reponse</TemplateName>
<TemplateType>InboundXSL</TemplateType>
<Description>This is the create ticket response template. </Description>
</TemplateRegistration>
<TemplateRegistration>
<FileName>templates/Remedy_DefaultCategory.xsl</FileName>
<InternalName>Remedy_DefaultCategory.xsl</InternalName>
<TemplateName>Remedy Default Category </TemplateName>
<TemplateType>OutboundXSL</TemplateType>
<Description>This is the Remdy default category template. </Description>
</TemplateRegistration>
<TemplateRegistration>
<FileName>templates/Remedy_DefaultCategory_AutoClose.xsl</FileName>
<InternalName>Remedy_DefaultCategory_AutoClose.xsl</InternalName>
<TemplateName>Remedy Default Category Auto Close</TemplateName>
<TemplateType>OutboundXSL</TemplateType>
<Description>This is the Remdy default category template with autp close function.
</Description>
</TemplateRegistration>
<TemplateRegistration>
<FileName>templates/Remedy_DefaultCategory_AutoResolve.xsl</FileName>
<InternalName>Remedy_DefaultCategory_AutoResolve.xsl</InternalName>
<TemplateName>Remedy Default Category Auto Resolve</TemplateName>
<TemplateType>OutboundXSL</TemplateType>
<Description>This is the Remdy default category template with autp resolve
function. </Description>
</TemplateRegistration>
<TemplateRegistration>
<FileName>publishTicket_request.xsl</FileName>
<InternalName>publishTicket</InternalName>
<TemplateName>Publish Ticket Status</TemplateName>
<TemplateType>InboundXSL</TemplateType>
<Description>This is the template for publishTicket operation. </Description>
</TemplateRegistration>
</ManagementConnector>
```

Example A–2 getTicket_request.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<urn:Get_By_IncidentId xmlns:urn="urn:HPD_IncidentInterface_get_WS">
<urn:Incident_Number>@TicketId@</urn:Incident_Number>
</urn:Get_By_IncidentId>
```

Example A–3 getTicket_response.xsl

```
<?xml version='1.0' encoding='UTF-8'?>
<xsl:transform version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:urn="urn:HPD_IncidentInterface_get_WS"
xmlns="http://xmlns.oracle.com/sysman/connector/tt"
targetNamespace="http://xmlns.oracle.com/sysman/connector/tt"
elementFormDefault="qualified">
<xsl:template match="urn:Get_By_IncidentIdResponse">
<getTicketResponse>
<TicketId><xsl:value-of select="urn:Incident_Number/text()"/></TicketId>
</getTicketResponse>
```

```
</xsl:template>
</xsl:transform>
```

Example A–4 createTicket_response.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:urn="urn:HPD_IncidentInterface_Create_WS"
xmlns="http://xmlns.oracle.com/sysman/connector">
<xsl:template match="urn:HelpDesk_Submit_ServiceResponse">
<CreateTicketResponse>
<TicketId>
<xsl:value-of select="urn:Incident_Number"/>
</TicketId>
<InstanceVariable>
<VariableName>Incident_Number</VariableName>
<VariableValue>
<xsl:value-of select="urn:Incident_Number"/>
</VariableValue>
</InstanceVariable>
</CreateTicketResponse>
</xsl:template>
</xsl:transform>
```

Example A–5 Remedy_DefaultCategory.xsl

```
<?xml version='1.0' encoding='UTF-8'?>
<xsl:transform version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:ns0="http://xmlns.oracle.com/sysman/connector"
targetNamespace="http://xmlns.oracle.com/sysman/connector"
elementFormDefault="qualified">

<!--This template creates an incident type ticket with default categorization.
The ticket priority is based on event severity. On update,the ticket summary
is updated with the latest incident message and severity information. -->
<xsl:template match="ns0:EMIncident">
<xsl:choose>
<xsl:when test="normalize-space(ns0:TicketID) = ''">
<urn:HelpDesk_Submit_Service xmlns:urn="urn:HPD_IncidentInterface_Create_WS">
<!-- EDIT THE TAG VALUES BELOW TO CHANGE HOW A TICKET IS FILLED DURING
TICKET CREATION. REFER TO THE REMEDY SERVICE DESK MANUAL FOR DESCRIPTION
OF THESE SERVICEDESK SUPPORT DATAFIELDS-->
<urn:Assigned_Group/>
<urn:Assigned_Group_Shift_Name/>
<urn:Assigned_Support_Company/>
<urn:Assigned_Support_Organization></urn:Assigned_Support_Organization>
<urn:Assignee/>
<urn:Categorization_Tier_1/>
<urn:Categorization_Tier_2/>
<urn:Categorization_Tier_3/>
<urn:CI_Name/>
<urn:Closure_Manufacturer/>
<urn:Closure_Product_Category_Tier1/>
<urn:Closure_Product_Category_Tier2/>
<urn:Closure_Product_Category_Tier3/>
<urn:Closure_Product_Model_Version/>
<urn:Closure_Product_Name/>
<urn:Department/>
<!--FIRST_NAME,LAST_NAME VALUES ARE PICKED FROM THE USERNAME VALUE GIVEN DURING
REMEDY SERVICE DESK CONNECTOR CONFIGURATION. EXAMPLE USERNAME:Demo.-->
```

```
<urn:First_Name><xsl:value-of select="ns0:HDUser"/></urn:First_Name>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'FATAL')">
<urn:Impact>1-Extensive/Widespread</urn:Impact>
</xsl:when>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'CRITICAL')">
<urn:Impact>2-Significant/Large</urn:Impact>
</xsl:when>
<xsl:otherwise>
<urn:Impact>3-Moderate/Limited</urn:Impact>
</xsl:otherwise>
</xsl:choose>
<urn:Last_Name><xsl:value-of select="ns0:HDUser"/></urn:Last_Name>
<urn:Lookup_Keyword/>
<urn:Manufacturer/>
<urn:Product_Categorization_Tier_1/>
<urn:Product_Categorization_Tier_2/>
<urn:Product_Categorization_Tier_3/>
<urn:Product_Model_Version/>
<urn:Product_Name/>
<urn:Reported_Source>Systems Management</urn:Reported_Source>
<urn:Resolution/>
<urn:Resolution_Category_Tier_1/>
<urn:Resolution_Category_Tier_2/>
<urn:Resolution_Category_Tier_3/>
<urn:Service_Type>Infrastructure Event</urn:Service_Type>
<urn:Status>New</urn:Status>
<urn:Action>CREATE</urn:Action>
<urn:Create_Request/>
<urn:Summary>
<xsl:value-of select="substring(ns0:SystemAttributes/ns0:Summary,0, 100)"/>
</urn:Summary>
<urn:Notes>
Incident created by Oracle Enterprise Manager Remedy Service Desk Connector.
-----------------------------------------------------------------------------
EM User: <xsl:value-of select="ns0:NotificationRuleOwner"/>
Incident Information:
<xsl:for-each select="ns0:SystemAttributes/ns0:SourceInfo">
Source Name:<xsl:value-of select="./ns0:SourceObjInfo/ns0:ObjName" />
<xsl:choose>
<xsl:when test="normalize-space(./ns0:SourceObjInfo/ns0:ObjOwner) != ''">
Source Owner:<xsl:value-of select="./ns0:SourceObjInfo/ns0:ObjOwner" />
</xsl:when>
</xsl:choose>
Source Type:<xsl:value-of select="./ns0:SourceObjInfo/ns0:SourceObjType"/>
Source SubType:<xsl:value-of select="./ns0:SourceObjInfo/ns0:SourceObjSubType"/>
Target Name: <xsl:value-of select="./ns0:TargetInfo/ns0:TargetName"/>
Target Type: <xsl:value-of select="./ns0:TargetInfo/ns0:TargetType"/>
Target Type Label: <xsl:value-of select="./ns0:TargetInfo/ns0:TargetTypeLabel"/>
Target URL:<xsl:value-of select="./ns0:TargetInfo/ns0:TargetURL"/>
<xsl:text>&#xa; </xsl:text>
</xsl:for-each>
<!-- LIST ALL THE TARGET PROPERTIES -->
Target Properties:
<xsl:for-each select="ns0:SystemAttributes/ns0:SourceInfo/ns0:TargetInfo/
ns0:TargetProperty">
<xsl:text>&#xa; </xsl:text>
<xsl:value-of select="./ns0:Name"/>: <xsl:value-of select="./ns0:Value"/>
</xsl:for-each>
<!-- EDIT THE FOLLOWING CODE TO LIST A SPECIFIC TARGET PROPERTY,
SUCH AS "Line of Business"
<xsl:choose>
<xsl:when test="ns0:SystemAttributes/ns0:SourceInfo/ns0:TargetInfo/
ns0:TargetProperty/ns0:Name='Line of Business'">
Line of Business: <xsl:value-of select="ns0:ns0:SystemAttributes/
ns0:SourceInfo/ns0:TargetInfo/ns0:TargetProperty
/ns0:value"/>
</xsl:when>
</xsl:choose>
-->
Severity: <xsl:value-of select="ns0:SystemAttributes/ns0:Severity"/>
Priority: <xsl:value-of select="ns0:SystemAttributes/ns0:Priority" />
CreationDate: <xsl:value-of select="ns0:SystemAttributes/ns0:CreationDate"/>
LastUpdatedDate:<xsl:value-of select="ns0:SystemAttributes/ns0:LastUpdatedDate"/>
Owner: <xsl:value-of select="ns0:SystemAttributes/ns0:Owner" />
<xsl:choose>
<xsl:when test="normalize-space(ns0:NotificationRuleName) != ''">
Notification Rule: <xsl:value-of select="ns0:NotificationRuleName"/>
</xsl:when>
</xsl:choose>
URL: <xsl:value-of select="ns0:SystemAttributes/ns0:IncidentURL"/>
</urn:Notes>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'FATAL') or
(ns0:SystemAttributes/ns0:SeverityCode = 'CRITICAL') ">
<urn:Urgency>1-Critical</urn:Urgency>
</xsl:when>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'WARNING')">
<urn:Urgency>2-High</urn:Urgency>
</xsl:when>
<xsl:otherwise>
<urn:Urgency>3-Medium</urn:Urgency>
</xsl:otherwise>
</xsl:choose>
<urn:Work_Info_Summary>
<xsl:value-of select="substring(ns0:SystemAttributes/ns0:Summary, 0 , 100)"/>
</urn:Work_Info_Summary>
<urn:Work_Info_Notes>
Incident created by Oracle Enterprise Manager Remedy Service Desk Connector for
<xsl:value-of select="ns0:SystemAttributes/ns0:Severity"/> severity.
</urn:Work_Info_Notes>
<urn:Work_Info_Type>Incident Task / Action</urn:Work_Info_Type>
<urn:Work_Info_Date/>
<urn:Work_Info_Source>System Assignment</urn:Work_Info_Source>
<urn:Work_Info_Locked/>
<urn:Work_Info_View_Access>Public</urn:Work_Info_View_Access>
<urn:Middle_Initial/>
<urn:ConnectorGUID>
<xsl:value-of select="ns0:ConnectorGUID"/>
</urn:ConnectorGUID>
</urn:HelpDesk_Submit_Service>
</xsl:when>
<xsl:otherwise>
<urn:HelpDesk_Modify_Service xmlns:urn="urn:HPD_IncidentInterface_WS">
<urn:Categorization_Tier_1></urn:Categorization_Tier_1>
<urn:Categorization_Tier_2></urn:Categorization_Tier_2>
<urn:Categorization_Tier_3></urn:Categorization_Tier_3>
<urn:Closure_Manufacturer></urn:Closure_Manufacturer>
<urn:Closure_Product_Category_Tier1></urn:Closure_Product_Category_Tier1>
<urn:Closure_Product_Category_Tier2></urn:Closure_Product_Category_Tier2>
```

```
<urn:Closure_Product_Category_Tier3></urn:Closure_Product_Category_Tier3>
<urn:Closure_Product_Model_Version></urn:Closure_Product_Model_Version>
<urn:Closure_Product_Name></urn:Closure_Product_Name>
<!--EDIT THE Company TAG BELOW TO ADD A Company NAME THAT IS ASSOCIATED WITH
FIRST_NAME,LAST_NAME TAGS ON THE REMEDY -->
<urn:Company>My Company</urn:Company>
<urn:Summary>
<xsl:value-of select="substring(ns0:SystemAttributes/ns0:Summary, 0 , 100)"/>
</urn:Summary>
<urn:Notes>
Incident updated by Oracle Enterprise Manager Remedy Service Desk Connector.
---------------------------------------------------------------------------
EM User: <xsl:value-of select="ns0:NotificationRuleOwner"/>
Incident Information:

Source Name:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:ObjName" />
<xsl:choose>
<xsl:when test="normalize-space(ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:ObjOwner) != ''">
Source Owner:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:ObjOwner" />
</xsl:when>
</xsl:choose>
Source Type:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:SourceObjType"/>
Source SubType:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:SourceObjSubType"/>

Target Name: <xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetName"/>
Target Type: <xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetType"/>
Target Type Label: <xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetTypeLabel"/>
Target URL:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetURL"/>
<!-- LIST ALL THE TARGET PROPERTIES -->
Target Properties:
<xsl:for-each select="ns0:SystemAttributes/ns0:SourceInfo/ns0:TargetInfo/
ns0:TargetProperty">
<xsl:text>&#xa; </xsl:text>
<xsl:value-of select="./ns0:Name"/>: <xsl:value-of select="./ns0:Value"/>
</xsl:for-each>
<!-- EDIT THE FOLLOWING CODE TO LIST A SPECIFIC TARGET PROPERTY,
SUCH AS "Line of Business"
<xsl:choose>
<xsl:when test="ns0:SystemAttributes/ns0:SourceInfo/ns0:TargetInfo/
ns0:TargetProperty/ns0:Name='Line of Business'">
Line of Business: <xsl:value-of select="ns0:ns0:SystemAttributes/
ns0:SourceInfo/ns0:TargetInfo/ns0:TargetProperty
/ns0:value"/>
</xsl:when>
</xsl:choose>
-->
Severity: <xsl:value-of select="ns0:SystemAttributes/ns0:Severity"/>
Priority: <xsl:value-of select="ns0:SystemAttributes/ns0:Priority" />
CreationDate: <xsl:value-of select="ns0:SystemAttributes/ns0:CreationDate"/>
LastUpdatedDate:<xsl:value-of select="ns0:SystemAttributes/ns0:LastUpdatedDate"/>
Owner: <xsl:value-of select="ns0:SystemAttributes/ns0:Owner" />
```

```
<xsl:choose>
<xsl:when test="normalize-space(ns0:NotificationRuleName) != ''">
Notification Rule: <xsl:value-of select="ns0:NotificationRuleName"/>
</xsl:when>
</xsl:choose>
URL: <xsl:value-of select="ns0:SystemAttributes/ns0:IncidentURL"/>
</urn:Notes>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'FATAL')">
<urn:Impact>1-Extensive/Widespread</urn:Impact>
</xsl:when>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'CRITICAL')">
<urn:Impact>2-Significant/Large</urn:Impact>
</xsl:when>
<xsl:otherwise>
<urn:Impact>3-Moderate/Limited</urn:Impact>
</xsl:otherwise>
</xsl:choose>
<urn:Manufacturer></urn:Manufacturer>
<urn:Product_Categorization_Tier_1></urn:Product_Categorization_Tier_1>
<urn:Product_Categorization_Tier_2></urn:Product_Categorization_Tier_2>
<urn:Product_Categorization_Tier_3></urn:Product_Categorization_Tier_3>
<urn:Product_Model_Version></urn:Product_Model_Version>
<urn:Product_Name></urn:Product_Name>
<urn:Reported_Source>Systems Management</urn:Reported_Source>
<urn:Resolution></urn:Resolution>
<urn:Resolution_Category/>
<urn:Resolution_Category_Tier_2/>
<urn:Resolution_Category_Tier_3/>
<urn:Resolution_Method/>
<urn:Service_Type>Infrastructure Event</urn:Service_Type>
<urn:Status>Assigned</urn:Status>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'FATAL') or
(ns0:SystemAttributes/ns0:SeverityCode = 'CRITICAL') ">
<urn:Urgency>1-Critical</urn:Urgency>
</xsl:when>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'WARNING')">
<urn:Urgency>2-High</urn:Urgency>
</xsl:when>
<xsl:otherwise>
<urn:Urgency>3-Medium</urn:Urgency>
</xsl:otherwise>
</xsl:choose>
<urn:Action>MODIFY</urn:Action>
<urn:Work_Info_Summary>
<xsl:value-of select="substring(ns0:SystemAttributes/ns0:Summary, 0, 100)"/>
</urn:Work_Info_Summary>
<urn:Work_Info_Notes>Incident updated due to change in associated Incident:
<xsl:value-of select="ns0:SystemAttributes/ns0:Severity"/> severity.
</urn:Work_Info_Notes>
<urn:Work_Info_Type>Incident Task / Action</urn:Work_Info_Type>
<urn:Work_Info_Date/>
<urn:Work_Info_Source>System Assignment</urn:Work_Info_Source>
<urn:Work_Info_Locked>No</urn:Work_Info_Locked>
<urn:Work_Info_View_Access>Public</urn:Work_Info_View_Access>
<urn:Incident_Number>
<xsl:value-of select="ns0:TicketID"/>
</urn:Incident_Number>
<urn:ConnectorGUID>
```

```
<xsl:value-of select="ns0:ConnectorGUID"/>
</urn:ConnectorGUID>
</urn:HelpDesk_Modify_Service>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
</xsl:transform>
```

Example A–6 Remedy_DefaultAutoClose.xsl

```
<?xml version='1.0' encoding='UTF-8'?>
<xsl:transform version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:ns0="http://xmlns.oracle.com/sysman/connector"
targetNamespace="http://xmlns.oracle.com/sysman/connector"
elementFormDefault="qualified">

<!--
This template creates an incident type ticket within Remedy Service Desk with
default settings. On update, the worklog is updated with the latest event
message and severity information. The template supports auto closing of
tickets, once the ticket is closed it can not be reopened.
-->
<xsl:template match="ns0:EMIncident">
<xsl:choose>
<xsl:when test="normalize-space(ns0:TicketID) = ''" >
<urn:HelpDesk_Submit_Service xmlns:urn="urn:HPD_IncidentInterface_Create_WS">
<!-- EDIT THE TAG VALUES BELOW TO CHANGE HOW A TICKET IS FILLED
DURING TICKET CREATION. REFER TO THE REMEDY SERVICE DESK MANUAL
FOR DESCRIPTION OF THESE HELPDESK SUPPORT DATAFIELDS-->
<urn:Assigned_Group/>
<urn:Assigned_Group_Shift_Name/>
<urn:Assigned_Support_Company/>
<urn:Assigned_Support_Organization></urn:Assigned_Support_Organization>
<urn:Assignee/>
<urn:Categorization_Tier_1/>
<urn:Categorization_Tier_2/>
<urn:Categorization_Tier_3/>
<urn:CI_Name/>
<urn:Closure_Manufacturer/>
<urn:Closure_Product_Category_Tier1/>
<urn:Closure_Product_Category_Tier2/>
<urn:Closure_Product_Category_Tier3/>
<urn:Closure_Product_Model_Version/>
<urn:Closure_Product_Name/>
<urn:Department/>
<!--FIRST_NAME,LAST_NAME VALUES ARE PICKED FROM THE USERNAME VALUE GIVEN DURING
REMEDY SERVICE DESK CONNECTOR CONFIGURATION. EXAMPLE USERNAME:Demo.-->
<urn:First_Name><xsl:value-of select="ns0:HDUser"/></urn:First_Name>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'FATAL')">
<urn:Impact>1-Extensive/Widespread</urn:Impact>
</xsl:when>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'CRITICAL')">
<urn:Impact>2-Significant/Large</urn:Impact>
</xsl:when>
<xsl:otherwise>
<urn:Impact>3-Moderate/Limited</urn:Impact>
</xsl:otherwise>
</xsl:choose>
```

```
<urn:Last_Name><xsl:value-of select="ns0:HDUser"/></urn:Last_Name>
<urn:Lookup_Keyword/>
<urn:Manufacturer/>
<urn:Product_Categorization_Tier_1/>
<urn:Product_Categorization_Tier_2/>
<urn:Product_Categorization_Tier_3/>
<urn:Product_Model_Version/>
<urn:Product_Name/>
<urn:Reported_Source>Systems Management</urn:Reported_Source>
<urn:Resolution/>
<urn:Resolution_Category_Tier_1/>
<urn:Resolution_Category_Tier_2/>
<urn:Resolution_Category_Tier_3/>
<urn:Service_Type>Infrastructure Event</urn:Service_Type>
<urn:Status>New</urn:Status>
<urn:Action>CREATE</urn:Action>
<urn:Create_Request/>
<urn:Summary>
<xsl:value-of select="substring(ns0:SystemAttributes/ns0:Summary, 0, 100)"/>
</urn:Summary>
<urn:Notes>
Incident created by Oracle Enterprise Manager Remedy Service Desk Connector.
-----------------------------------------------------------------------------
EM User: <xsl:value-of select="ns0:NotificationRuleOwner"/>
Incident Information:

Source Name:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:ObjName" />
<xsl:choose>
<xsl:when test="normalize-space(ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:ObjOwner) != ''">
Source Owner:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:ObjOwner" />
</xsl:when>
</xsl:choose>
Source Type:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:SourceObjType"/>
Source SubType:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:SourceObjSubType"/>

Target Name: <xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetName"/>
Target Type: <xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetType"/>
Target Type Label: <xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetTypeLabel"/>
Target URL:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetURL"/>
<!-- LIST ALL THE TARGET PROPERTIES -->
Target Properties:
<xsl:for-each select="ns0:SystemAttributes/ns0:SourceInfo/ns0:TargetInfo/
ns0:TargetProperty">
<xsl:text>&#xa; </xsl:text>
<xsl:value-of select="./ns0:Name"/>: <xsl:value-of select="./ns0:Value"/>
</xsl:for-each>
<!-- EDIT THE FOLLOWING CODE TO LIST A SPECIFIC TARGET PROPERTY,
SUCH AS "Line of Business"
<xsl:choose>
<xsl:when test="ns0:SystemAttributes/ns0:SourceInfo/ns0:TargetInfo/
ns0:TargetProperty/ns0:Name='Line of Business'">
```

```
Line of Business: <xsl:value-of select="ns0:ns0:SystemAttributes/
ns0:SourceInfo/ns0:TargetInfo/ns0:TargetProperty
/ns0:value"/>
</xsl:when>
</xsl:choose>
-->
Severity: <xsl:value-of select="ns0:SystemAttributes/ns0:Severity"/>
Priority: <xsl:value-of select="ns0:SystemAttributes/ns0:Priority" />
CreationDate: <xsl:value-of select="ns0:SystemAttributes/ns0:CreationDate"/>
LastUpdatedDate:<xsl:value-of select="ns0:SystemAttributes/ns0:LastUpdatedDate"/>
Owner: <xsl:value-of select="ns0:SystemAttributes/ns0:Owner" />
<xsl:choose>
<xsl:when test="normalize-space(ns0:NotificationRuleName) != ''">
Notification Rule: <xsl:value-of select="ns0:NotificationRuleName"/>
</xsl:when>
</xsl:choose>
URL: <xsl:value-of select="ns0:SystemAttributes/ns0:IncidentURL"/>
</urn:Notes>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'FATAL') or
(ns0:SystemAttributes/ns0:SeverityCode = 'CRITICAL') ">
<urn:Urgency>1-Critical</urn:Urgency>
</xsl:when>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'WARNING')">
<urn:Urgency>2-High</urn:Urgency>
</xsl:when>
<xsl:otherwise>
<urn:Urgency>3-Medium</urn:Urgency>
</xsl:otherwise>
</xsl:choose>
<urn:Work_Info_Summary>
<xsl:value-of select="substring(ns0:SystemAttributes/ns0:Summary, 0, 100)"/>
</urn:Work_Info_Summary>
<urn:Work_Info_Notes>
Incident created by Oracle Enterprise Manager Remedy Service Desk Connector for
<xsl:value-of select="ns0:SystemAttributes/ns0:Severity"/> severity.
</urn:Work_Info_Notes>
<urn:Work_Info_Type>Incident Task / Action</urn:Work_Info_Type>
<urn:Work_Info_Date/>
<urn:Work_Info_Source>System Assignment</urn:Work_Info_Source>
<urn:Work_Info_Locked/>
<urn:Work_Info_View_Access>Public</urn:Work_Info_View_Access>
<urn:Middle_Initial/>
<urn:ConnectorGUID>
<xsl:value-of select="ns0:ConnectorGUID"/>
</urn:ConnectorGUID>
</urn:HelpDesk_Submit_Service>
</xsl:when>
<xsl:otherwise>
<urn:HelpDesk_Modify_Service xmlns:urn="urn:HPD_IncidentInterface_WS">
<urn:Categorization_Tier_1></urn:Categorization_Tier_1>
<urn:Categorization_Tier_2></urn:Categorization_Tier_2>
<urn:Categorization_Tier_3></urn:Categorization_Tier_3>
<urn:Closure_Manufacturer></urn:Closure_Manufacturer>
<urn:Closure_Product_Category_Tier1></urn:Closure_Product_Category_Tier1>
<urn:Closure_Product_Category_Tier2></urn:Closure_Product_Category_Tier2>
<urn:Closure_Product_Category_Tier3></urn:Closure_Product_Category_Tier3>
<urn:Closure_Product_Model_Version></urn:Closure_Product_Model_Version>
<urn:Closure_Product_Name></urn:Closure_Product_Name>
<!--EDIT THE Company TAG BELOW TO ADD A Company NAME THAT IS ASSOCIATED WITH
```

```
FIRST_NAME,LAST_NAME TAGS ON THE REMEDY -->
<urn:Company>My Company</urn:Company>
<urn:Summary>
<xsl:value-of select="substring(ns0:SystemAttributes/ns0:Summary, 0, 100)"/>
</urn:Summary>
<urn:Notes>
Incident updated by Oracle Enterprise Manager Remedy Service Desk Connector.
------------------------------------------------------------------------
EM User: <xsl:value-of select="ns0:NotificationRuleOwner"/>
Incident Information:

Source Name:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:ObjName" />
<xsl:choose>
<xsl:when test="normalize-space(ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:ObjOwner) != ''">
Source Owner:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:ObjOwner" />
</xsl:when>
</xsl:choose>
Source Type:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:SourceObjType"/>
Source SubType:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:SourceObjSubType"/>

Target Name: <xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetName"/>
Target Type: <xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetType"/>
Target Type Label: <xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetTypeLabel"/>
Target URL:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetURL"/>
<!-- LIST ALL THE TARGET PROPERTIES -->
Target Properties:
<xsl:for-each select="ns0:SystemAttributes/ns0:SourceInfo/ns0:TargetInfo/
ns0:TargetProperty">
<xsl:text>&#xa; </xsl:text>
<xsl:value-of select="./ns0:Name"/>: <xsl:value-of select="./ns0:Value"/>
</xsl:for-each>
<!-- EDIT THE FOLLOWING CODE TO LIST A SPECIFIC TARGET PROPERTY,
SUCH AS "Line of Business"
<xsl:choose>
<xsl:when test="ns0:SystemAttributes/ns0:SourceInfo/ns0:TargetInfo/
ns0:TargetProperty/ns0:Name='Line of Business'">
Line of Business: <xsl:value-of select="ns0:ns0:SystemAttributes/
ns0:SourceInfo/ns0:TargetInfo/ns0:TargetProperty
/ns0:value"/>
</xsl:when>
</xsl:choose>
-->
Severity: <xsl:value-of select="ns0:SystemAttributes/ns0:Severity"/>
Priority: <xsl:value-of select="ns0:SystemAttributes/ns0:Priority" />
CreationDate: <xsl:value-of select="ns0:SystemAttributes/ns0:CreationDate"/>
LastUpdatedDate:<xsl:value-of select="ns0:SystemAttributes/ns0:LastUpdatedDate"/>
Owner: <xsl:value-of select="ns0:SystemAttributes/ns0:Owner" />
<xsl:choose>
<xsl:when test="normalize-space(ns0:NotificationRuleName) != ''">
Notification Rule: <xsl:value-of select="ns0:NotificationRuleName"/>
</xsl:when>
```

```
</xsl:choose>
URL: <xsl:value-of select="ns0:SystemAttributes/ns0:IncidentURL"/>
</urn:Notes>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'FATAL')">
<urn:Impact>1-Extensive/Widespread</urn:Impact>
</xsl:when>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'CRITICAL')">
<urn:Impact>2-Significant/Large</urn:Impact>
</xsl:when>
<xsl:otherwise>
<urn:Impact>3-Moderate/Limited</urn:Impact>
</xsl:otherwise>
</xsl:choose>
<urn:Manufacturer></urn:Manufacturer>
<urn:Product_Categorization_Tier_1></urn:Product_Categorization_Tier_1>
<urn:Product_Categorization_Tier_2></urn:Product_Categorization_Tier_2>
<urn:Product_Categorization_Tier_3></urn:Product_Categorization_Tier_3>
<urn:Product_Model_Version></urn:Product_Model_Version>
<urn:Product_Name></urn:Product_Name>
<urn:Reported_Source>Systems Management</urn:Reported_Source>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'CLEAR')">
<urn:Resolution>
Incident closed by Oracle Enterprise Manager Remedy
Service Desk Connector due to change in severity of the
associated alert. Severity:
<xsl:value-of select="ns0:SystemAttributes/ns0:Severity"/>
Message:
<xsl:value-of select="substring(ns0:SystemAttributes/ns0:Summary, 0, 100)"/>
</urn:Resolution>
</xsl:when>
<xsl:otherwise>
<urn:Resolution></urn:Resolution>
</xsl:otherwise>
</xsl:choose>
<urn:Resolution_Category/>
<urn:Resolution_Category_Tier_2/>
<urn:Resolution_Category_Tier_3/>
<urn:Resolution_Method/>
<urn:Service_Type>Infrastructure Event</urn:Service_Type>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'CLEAR')">
<urn:Status>Closed</urn:Status>
</xsl:when>
<xsl:otherwise>
<urn:Status>Assigned</urn:Status>
</xsl:otherwise>
</xsl:choose>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'FATAL') or
(ns0:SystemAttributes/ns0:SeverityCode = 'CRITICAL') ">
<urn:Urgency>1-Critical</urn:Urgency>
</xsl:when>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'WARNING')">
<urn:Urgency>2-High</urn:Urgency>
</xsl:when>
<xsl:otherwise>
<urn:Urgency>3-Medium</urn:Urgency>
</xsl:otherwise>
```

```
</xsl:choose>
<urn:Action>MODIFY</urn:Action>
<urn:Work_Info_Summary>
<xsl:value-of select="substring(ns0:SystemAttributes/ns0:Summary, 0, 100)"/>
</urn:Work_Info_Summary>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'CLEAR')">
<urn:Work_Info_Notes>
Incident closed by Oracle Enterprise Manager Remedy Service Desk Connector due to
change in associated Incident: <xsl:value-of
select="ns0:SystemAttributes/ns0:Severity"/> severity.
</urn:Work_Info_Notes>
</xsl:when>
<xsl:otherwise>
<urn:Work_Info_Notes>Incident updated due to change in associated Incident:
<xsl:value-of select="ns0:SystemAttributes/ns0:Severity"/> severity.
</urn:Work_Info_Notes>
</xsl:otherwise>
</xsl:choose>
<urn:Work_Info_Type>Incident Task / Action</urn:Work_Info_Type>
<urn:Work_Info_Date/>
<urn:Work_Info_Source>System Assignment</urn:Work_Info_Source>
<urn:Work_Info_Locked>No</urn:Work_Info_Locked>
<urn:Work_Info_View_Access>Public</urn:Work_Info_View_Access>
<urn:Incident_Number>
<xsl:value-of select="ns0:TicketID"/>
</urn:Incident_Number>
<urn:ConnectorGUID>
<xsl:value-of select="ns0:ConnectorGUID"/>
</urn:ConnectorGUID>
</urn:HelpDesk_Modify_Service>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
</xsl:transform>
```

Example A–7 Remedy_DefaultAutoResolve.xsl

```
<?xml version='1.0' encoding='UTF-8'?>
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:ns0="http://xmlns.oracle.com/sysman/connector"
targetNamespace="http://xmlns.oracle.com/sysman/connector"
elementFormDefault="qualified">
<!-- This template creates an incident type ticket within Remedy Service
Desk with default settings. On update, the worklog is updated with the latest
incident message and severity information. The ticket is set to status Resolved
if the associated alert has cleared. Ticket can be reopend if a severity
occured with in the grace period. If the ticket is not reopened for 15 days,
ticket will be closed by incident management.
-->
<xsl:template match="ns0:EMIncident">
<xsl:choose>
<xsl:when test="normalize-space(ns0:TicketID) = ''">
<urn:HelpDesk_Submit_Service xmlns:urn="urn:HPD_IncidentInterface_Create_WS">
<!-- EDIT THE TAG VALUES BELOW TO CHANGE HOW A TICKET IS FILLED
DURING TICKET CREATION. REFER TO THE REMEDY SERVICE DESK MANUAL
FOR DESCRIPTION OF THESE HELPDESK SUPPORT DATAFIELDS-->
<urn:Assigned_Group/>
<urn:Assigned_Group_Shift_Name/>
<urn:Assigned_Support_Company/>
```

```
<urn:Assigned_Support_Organization></urn:Assigned_Support_Organization>
<urn:Assignee/>
<urn:Categorization_Tier_1/>
<urn:Categorization_Tier_2/>
<urn:Categorization_Tier_3/>
<urn:CI_Name/>
<urn:Closure_Manufacturer/>
<urn:Closure_Product_Category_Tier1/>
<urn:Closure_Product_Category_Tier2/>
<urn:Closure_Product_Category_Tier3/>
<urn:Closure_Product_Model_Version/>
<urn:Closure_Product_Name/>
<urn:Department/>
<!--FIRST_NAME,LAST_NAME VALUES ARE PICKED FROM THE USERNAME VALUE GIVEN DURING
REMEDY SERVICE DESK CONNECTOR CONFIGURATION. EXAMPLE USERNAME:Demo.-->
<urn:First_Name><xsl:value-of select="ns0:HDUser"/></urn:First_Name>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'FATAL')">
<urn:Impact>1-Extensive/Widespread</urn:Impact>
</xsl:when>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'CRITICAL')">
<urn:Impact>2-Significant/Large</urn:Impact>
</xsl:when>
<xsl:otherwise>
<urn:Impact>3-Moderate/Limited</urn:Impact>
</xsl:otherwise>
</xsl:choose>
<urn:Last_Name><xsl:value-of select="ns0:HDUser"/></urn:Last_Name>
<urn:Lookup_Keyword/>
<urn:Manufacturer/>
<urn:Product_Categorization_Tier_1/>
<urn:Product_Categorization_Tier_2/>
<urn:Product_Categorization_Tier_3/>
<urn:Product_Model_Version/>
<urn:Product_Name/>
<urn:Reported_Source>Systems Management</urn:Reported_Source>
<urn:Resolution/>
<urn:Resolution_Category_Tier_1/>
<urn:Resolution_Category_Tier_2/>
<urn:Resolution_Category_Tier_3/>
<urn:Service_Type>Infrastructure Event</urn:Service_Type>
<urn:Status>New</urn:Status>
<urn:Action>CREATE</urn:Action>
<urn:Create_Request/>
<urn:Summary>
<xsl:value-of select="substring(ns0:SystemAttributes/ns0:Summary, 0, 100)"/>
</urn:Summary>
<urn:Notes>
Incident created by Oracle Enterprise Manager Remedy Service Desk Connector.
-------------------------------------------------------------------------------
EM User: <xsl:value-of select="ns0:NotificationRuleOwner"/>
Incident Information:

Source Name:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:ObjName" />
<xsl:choose>
<xsl:when test="normalize-space(ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:ObjOwner) != ''">
Source Owner:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:ObjOwner" />
```

```
</xsl:when>
</xsl:choose>
Source Type:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:SourceObjType"/>
Source SubType:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:SourceObjSubType"/>

Target Name: <xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetName"/>
Target Type: <xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetType"/>
Target Type Label: <xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetTypeLabel"/>
Target URL:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetURL"/>
<!-- LIST ALL THE TARGET PROPERTIES -->
Target Properties:
<xsl:for-each select="ns0:SystemAttributes/ns0:SourceInfo/ns0:TargetInfo/
ns0:TargetProperty">
<xsl:text>&#xa; </xsl:text>
<xsl:value-of select="./ns0:Name"/>: <xsl:value-of select="./ns0:Value"/>
</xsl:for-each>
<!-- EDIT THE FOLLOWING CODE TO LIST A SPECIFIC TARGET PROPERTY,
SUCH AS "Line of Business"
<xsl:choose>
<xsl:when test="ns0:SystemAttributes/ns0:SourceInfo/ns0:TargetInfo/
ns0:TargetProperty/ns0:Name='Line of Business'">
Line of Business: <xsl:value-of select="ns0:ns0:SystemAttributes/
ns0:SourceInfo/ns0:TargetInfo/ns0:TargetProperty
/ns0:value"/>
</xsl:when>
</xsl:choose>
-->
Severity: <xsl:value-of select="ns0:SystemAttributes/ns0:Severity"/>
Priority: <xsl:value-of select="ns0:SystemAttributes/ns0:Priority" />
CreationDate: <xsl:value-of select="ns0:SystemAttributes/ns0:CreationDate"/>
LastUpdatedDate:<xsl:value-of select="ns0:SystemAttributes/ns0:LastUpdatedDate"/>
Owner: <xsl:value-of select="ns0:SystemAttributes/ns0:Owner" />
<xsl:choose>
<xsl:when test="normalize-space(ns0:NotificationRuleName) != ''">
Notification Rule: <xsl:value-of select="ns0:NotificationRuleName"/>
</xsl:when>
</xsl:choose>
URL: <xsl:value-of select="ns0:SystemAttributes/ns0:IncidentURL"/>
</urn:Notes>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'FATAL') or
(ns0:SystemAttributes/ns0:SeverityCode = 'CRITICAL') ">
<urn:Urgency>1-Critical</urn:Urgency>
</xsl:when>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'WARNING')">
<urn:Urgency>2-High</urn:Urgency>
</xsl:when>
<xsl:otherwise>
<urn:Urgency>3-Medium</urn:Urgency>
</xsl:otherwise>
</xsl:choose>

<urn:Work_Info_Summary>
<xsl:value-of select="substring(ns0:SystemAttributes/ns0:Summary, 0, 100)"/>
```

```
</urn:Work_Info_Summary>
<urn:Work_Info_Notes>
Incident created by Oracle Enterprise Manager Remedy Service Desk Connector for
<xsl:value-of select="ns0:SystemAttributes/ns0:Severity"/> severity.
</urn:Work_Info_Notes>
<urn:Work_Info_Type>Incident Task / Action</urn:Work_Info_Type>
<urn:Work_Info_Date/>
<urn:Work_Info_Source>System Assignment</urn:Work_Info_Source>
<urn:Work_Info_Locked/>
<urn:Work_Info_View_Access>Public</urn:Work_Info_View_Access>
<urn:Middle_Initial/>
<urn:ConnectorGUID>
<xsl:value-of select="ns0:ConnectorGUID"/>
</urn:ConnectorGUID>
</urn:HelpDesk_Submit_Service>
</xsl:when>
<xsl:otherwise>
<urn:HelpDesk_Modify_Status_Service xmlns:urn="urn:HPD_IncidentInterface_WS">
<urn:Categorization_Tier_1></urn:Categorization_Tier_1>
<urn:Categorization_Tier_2></urn:Categorization_Tier_2>
<urn:Categorization_Tier_3></urn:Categorization_Tier_3>
<urn:Closure_Manufacturer></urn:Closure_Manufacturer>
<urn:Closure_Product_Category_Tier1></urn:Closure_Product_Category_Tier1>
<urn:Closure_Product_Category_Tier2></urn:Closure_Product_Category_Tier2>
<urn:Closure_Product_Category_Tier3></urn:Closure_Product_Category_Tier3>
<urn:Closure_Product_Model_Version></urn:Closure_Product_Model_Version>
<urn:Closure_Product_Name></urn:Closure_Product_Name>
<!--EDIT THE Company TAG BELOW TO ADD A Company NAME THAT IS ASSOCIATED WITH
FIRST_NAME,LAST_NAME TAGS ON THE REMEDY -->
<urn:Company>My Company</urn:Company>
<urn:Summary>
<xsl:value-of select="substring(ns0:SystemAttributes/ns0:Summary, 0, 100)"/>
</urn:Summary>
<urn:Notes>
Incident updated by Oracle Enterprise Manager Remedy Service Desk Connector.
-----------------------------------------------------------------------
EM User: <xsl:value-of select="ns0:NotificationRuleOwner"/>
Incident Information:

Source Name:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:ObjName" />
<xsl:choose>
<xsl:when test="normalize-space(ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:ObjOwner) != ''">
Source Owner:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:ObjOwner" />
</xsl:when>
</xsl:choose>
Source Type:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:SourceObjType"/>
Source SubType:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:SourceObjInfo/ns0:SourceObjSubType"/>

Target Name: <xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetName"/>
Target Type: <xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetType"/>
Target Type Label: <xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
ns0:TargetInfo/ns0:TargetTypeLabel"/>
Target URL:<xsl:value-of select="ns0:SystemAttributes/ns0:SourceInfo/
```

```
ns0:TargetInfo/ns0:TargetURL"/>
<!-- LIST ALL THE TARGET PROPERTIES -->
Target Properties:
<xsl:for-each select="ns0:SystemAttributes/ns0:SourceInfo/ns0:TargetInfo/
ns0:TargetProperty">
<xsl:text>&#xa; </xsl:text>
<xsl:value-of select="./ns0:Name"/>: <xsl:value-of select="./ns0:Value"/>
</xsl:for-each>
<!-- EDIT THE FOLLOWING CODE TO LIST A SPECIFIC TARGET PROPERTY,
SUCH AS "Line of Business"
<xsl:choose>
<xsl:when test="ns0:SystemAttributes/ns0:SourceInfo/ns0:TargetInfo/
ns0:TargetProperty/ns0:Name='Line of Business'">
Line of Business: <xsl:value-of select="ns0:ns0:SystemAttributes/
ns0:SourceInfo/ns0:TargetInfo/ns0:TargetProperty
/ns0:value"/>
</xsl:when>
</xsl:choose>
-->
Severity: <xsl:value-of select="ns0:SystemAttributes/ns0:Severity"/>
Priority: <xsl:value-of select="ns0:SystemAttributes/ns0:Priority" />
CreationDate: <xsl:value-of select="ns0:SystemAttributes/ns0:CreationDate"/>
LastUpdatedDate:<xsl:value-of select="ns0:SystemAttributes/ns0:LastUpdatedDate"/>
Owner: <xsl:value-of select="ns0:SystemAttributes/ns0:Owner" />
<xsl:choose>
<xsl:when test="normalize-space(ns0:NotificationRuleName) != ''">
Notification Rule: <xsl:value-of select="ns0:NotificationRuleName"/>
</xsl:when>
</xsl:choose>
URL: <xsl:value-of select="ns0:SystemAttributes/ns0:IncidentURL"/>
</urn:Notes>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'FATAL')">
<urn:Impact>1-Extensive/Widespread</urn:Impact>
</xsl:when>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'CRITICAL')">
<urn:Impact>2-Significant/Large</urn:Impact>
</xsl:when>
<xsl:otherwise>
<urn:Impact>3-Moderate/Limited</urn:Impact>
</xsl:otherwise>
</xsl:choose>
<urn:Incident_Number>
<xsl:value-of select="ns0:TicketID"/>
</urn:Incident_Number>
<urn:Manufacturer></urn:Manufacturer>
<urn:Product_Categorization_Tier_1></urn:Product_Categorization_Tier_1>
<urn:Product_Categorization_Tier_2></urn:Product_Categorization_Tier_2>
<urn:Product_Categorization_Tier_3></urn:Product_Categorization_Tier_3>
<urn:Product_Model_Version></urn:Product_Model_Version>
<urn:Product_Name></urn:Product_Name>
<urn:Reported_Source>Systems Management</urn:Reported_Source>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'CLEAR')">
<urn:Resolution>
Incident resolved by Oracle Enterprise Manager Remedy
Service Desk Connector due to change in severity of the
associated alert. Severity:
<xsl:value-of select="ns0:SystemAttributes/ns0:Severity"/>
Message:
```

```
<xsl:value-of select="substring(ns0:SystemAttributes/ns0:Summary, 0, 100)"/>
</urn:Resolution>
</xsl:when>
<xsl:otherwise>
<urn:Resolution></urn:Resolution>
</xsl:otherwise>
</xsl:choose>
<urn:Resolution_Category/>
<urn:Resolution_Category_Tier_2/>
<urn:Resolution_Category_Tier_3/>
<urn:Resolution_Method/>
<urn:Service_Type>Infrastructure Event</urn:Service_Type>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'CLEAR')">
<urn:Status>Resolved</urn:Status>
</xsl:when>
<xsl:otherwise>
<urn:Status>Assigned</urn:Status>
</xsl:otherwise>
</xsl:choose>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'CLEAR')">
<urn:Status_Reason>Automated Resolution Reported</urn:Status_Reason>
</xsl:when>
<xsl:otherwise>
<urn:Status_Reason></urn:Status_Reason>
</xsl:otherwise>
</xsl:choose>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'FATAL') or
(ns0:SystemAttributes/ns0:SeverityCode = 'CRITICAL') ">
<urn:Urgency>1-Critical</urn:Urgency>
</xsl:when>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'WARNING')">
<urn:Urgency>2-High</urn:Urgency>
</xsl:when>
<xsl:otherwise>
<urn:Urgency>3-Medium</urn:Urgency>
</xsl:otherwise>
</xsl:choose>
<urn:Action>MODIFY</urn:Action>
<urn:Work_Info_Type>Incident Task / Action</urn:Work_Info_Type>
<urn:Work_Info_Date/>
<urn:Work_Info_Source>System Assignment</urn:Work_Info_Source>
<xsl:choose>
<xsl:when test="(ns0:SystemAttributes/ns0:SeverityCode = 'CLEAR')">
<urn:Work_Info_Notes> Incident resolved by Oracle Enterprise Manager Remedy
Service Desk Connector due to change in associated Incident: <xsl:value-of
select="ns0:SystemAttributes/ns0:Severity"/> severity.
</urn:Work_Info_Notes>
</xsl:when>
<xsl:when test="ns0:ReopenTicket = 'Yes'">
<urn:Work_Info_Notes>Incident reopened because the associated alert re-triggered
within the grace period: <xsl:value-of
select="ns0:SystemAttributes/ns0:Severity"/> severity.
</urn:Work_Info_Notes>
</xsl:when>
<xsl:otherwise>
<urn:Work_Info_Notes>Incident updated due to change in associated Incident:
<xsl:value-of select="ns0:SystemAttributes/ns0:Severity"/> severity.
```

```
</urn:Work_Info_Notes>
</xsl:otherwise>
</xsl:choose>
<urn:Work_Info_Locked>No</urn:Work_Info_Locked>
<urn:Work_Info_View_Access>Public</urn:Work_Info_View_Access>
<urn:Work_Info_Summary>
<xsl:value-of select="substring(ns0:SystemAttributes/ns0:Summary, 0, 100)"/>
</urn:Work_Info_Summary>
<urn:ConnectorGUID>
<xsl:value-of select="ns0:ConnectorGUID"/>
</urn:ConnectorGUID>
</urn:HelpDesk_Modify_Status_Service>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
</xsl:transform>
```

Example A–8 connectorDeploy.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://xmlns.oracle.com/sysman/connector"
targetNamespace="http://xmlns.oracle.com/sysman/connector"
elementFormDefault="qualified">
<xsd:include schemaLocation="connectorCommon.xsd"/>
<xsd:element name="ManagementConnector">
<xsd:annotation>
<xsd:documentation>Deployment Descriptor for Management
Connectors</xsd:documentation>
</xsd:annotation>
<xsd:complexType>
<xsd:sequence>
<xsd:element name="Name" type="StringT64">
<xsd:annotation>
<xsd:documentation>
The name of the connector type.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Version" type="VersionT">
<xsd:annotation>
<xsd:documentation>
Version of the connector type.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="EMCompatibleVersion" type="VersionT">
<xsd:annotation>
<xsd:documentation>
The EM compabilitity version of the connector type.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Description" type="StringT256">
<xsd:annotation>
<xsd:documentation>
The description of the connector type.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
```

```
<xsd:element name="Category">
<xsd:annotation>
<xsd:documentation>
The category of the connector type. It must be one of the three
values listed next.
</xsd:documentation>
</xsd:annotation>
<xsd:simpleType>
<xsd:restriction base="xsd:string">
<xsd:enumeration value="EventConnector"/>
<xsd:enumeration value="TicketingConnector"/>
<xsd:enumeration value="ChangeManagementConnector"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
<!-- NewTargetType is for EventConnector only. -->
<xsd:element name="NewTargetType" minOccurs="0">
<xsd:annotation>
<xsd:documentation>
New target type definition for event connectors. This target type
will be registered with Enterprise Manager and target instances can
be created subsequently, including a default target. These targets
are used to accommodate external events.
</xsd:documentation>
</xsd:annotation>
<xsd:complexType>
<xsd:sequence>
<xsd:element name="TargetTypeName" type="StringStrictT64">
<xsd:annotation>
<xsd:documentation>
The name of the target type.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="TargetTypeDisplayName" type="StringT128">
<xsd:annotation>
<xsd:documentation>
The name of the target type, as shown on UI.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="DefaultTargetName" type="StringStrictT256">
<xsd:annotation>
<xsd:documentation>
The name of the default target of the target type. The default
target will be used as a generic bucket to hold external events.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="DefaultTargetDisplayName" type="StringT256">
<xsd:annotation>
<xsd:documentation>
The name of the default target of the target type, to be displayed
on UI.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
```

```
<xsd:element name="SOAPHeaderAuthentication"
type="SOAPHeaderAuthenticationType" minOccurs="0">
<xsd:annotation>
<xsd:documentation>
Specification for SOAP Header authentication.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="HTTPBasicAuthentication"
type="UsernamePasswordAuthenticationType" minOccurs="0">
<xsd:annotation>
<xsd:documentation>
Specification for HTTP basic authentication.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="UserNameTokenAuthentication"
type="UsernamePasswordAuthenticationType" minOccurs="0">
<xsd:annotation>
<xsd:documentation>
Specification for Username Token authentication.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="ConfigVariable" type="ConfigVariableType"
minOccurs="0" maxOccurs="20">
<xsd:annotation>
<xsd:documentation>
The vaiables used during connector configuration. These variables
are required by external system to complete connector configuration,
which includes regitering with the external system. For instance,
one configuration variable can be the resolution state required by
Microsoft Operation Manager.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="ConnectivityTestVariable" type="ConfigVariableType"
minOccurs="0">
<xsd:annotation>
<xsd:documentation>
An optional variable used to test connection to an external server.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Service" type="ServiceType" maxOccurs="20">
<xsd:annotation>
<xsd:documentation>
Specification for web services, which define how connector framework
can communicate with external system.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="ExternalURL" type="ExternalURLType" minOccurs="0">
<xsd:annotation>
<xsd:documentation>
Secification for the URL link to the external server, including
the URL pattern and server specific variables. It is used to provide links
to external server for viewing ticket details.
</xsd:documentation>
</xsd:annotation>
```

```
</xsd:element>
<xsd:element name="TemplateRegistration" type="TemplateRegistrationType"
minOccurs="0" maxOccurs="50">
<xsd:annotation>
<xsd:documentation>
Specification for template registration. A template is registered
based on the information provided in the element. A connector deployment
descriptor can have an optional list of upto 50 template registratin
elements.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:complexType name="ServiceType">
<xsd:annotation>
<xsd:documentation>
This section defines a complext type for a web service.
</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element name="Method">
<xsd:annotation>
<xsd:documentation>
The name of the web service method. Each connector category has a
predefined set of methods as defined next.
</xsd:documentation>
</xsd:annotation>
<xsd:simpleType>
<xsd:restriction base="xsd:string">
<!-- event connector: -->
<xsd:enumeration value="setup"/>
<xsd:enumeration value="initialize"/>
<xsd:enumeration value="getNewAlerts"/>
<xsd:enumeration value="getUpdatedAlerts"/>
<xsd:enumeration value="acknowledgeAlerts"/>
<xsd:enumeration value="updateAlerts"/>
<xsd:enumeration value="createEvent"/>
<xsd:enumeration value="updateEvent"/>
<xsd:enumeration value="uninitialize"/>
<xsd:enumeration value="cleanup"/>
<!-- ticketing connector: -->
<xsd:enumeration value="createTicket"/>
<xsd:enumeration value="updateTicket"/>
<xsd:enumeration value="getTicket"/>
<!-- change management connector: -->
<xsd:enumeration value="publishCS"/>
<xsd:enumeration value="updateChangeRequest"/>
<xsd:enumeration value="getChangeRequest"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="WebServiceEndpoint" type="StringT256">
<xsd:annotation>
<xsd:documentation>
The web service end point indicating a specific location for accessing
a service.
</xsd:documentation>
</xsd:annotation>
```

```
</xsd:element>
<xsd:element name="SOAPAction" type="StringT64" minOccurs="0">
<xsd:annotation>
<xsd:documentation>
The SOAP action which carries out the web service call for the method.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="SOAPBindingType" minOccurs="0">
<xsd:annotation>
<xsd:documentation>
The type of SOAP over HTTP binding. Choose from one of the four
options defined next.
</xsd:documentation>
</xsd:annotation>
<xsd:simpleType>
<xsd:restriction base="xsd:string">
<xsd:enumeration value="SOAP11HTTP_BINDING"/>
<xsd:enumeration value="SOAP12HTTP_BINDING"/>
<xsd:enumeration value="SOAP11HTTP_MTOM_BINDING"/>
<xsd:enumeration value="SOAP12HTTP_MTOM_BINDING"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SOAPHeaderAuthenticationType">
<xsd:annotation>
<xsd:documentation>
This section defines a complext type for SOAP Header Authentication.
</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element name="Username" type="ConfigVariableType">
<xsd:annotation>
<xsd:documentation>
The username of the authentication.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Password" type="ConfigVariableType">
<xsd:annotation>
<xsd:documentation>
The password of the authentication.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="AuthVariable" type="ConfigVariableType" minOccurs="0"
maxOccurs="20">
<xsd:annotation>
<xsd:documentation>
An optional list of extra authentication variables besides username
and password.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="SOAPHeader" type="StringT256">
<xsd:annotation>
<xsd:documentation>
A SOAP header string serving as template for the SOAP header. It is
```

```
to be updated with user inputs for variables defined above and
bound with a HTTP request.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="UsernamePasswordAuthenticationType">
<xsd:annotation>
<xsd:documentation>
This section defines a complext type for Username Password authentication.
</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element name="Username" type="ConfigVariableType">
<xsd:annotation>
<xsd:documentation>
The username of the authentication.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Password" type="ConfigVariableType">
<xsd:annotation>
<xsd:documentation>
The password of the authentication.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ConfigVariableType">
<xsd:annotation>
<xsd:documentation>
This seciton defines a complext type for configuration variables.
</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element name="VariableName" type="StringStrictT32">
<xsd:annotation>
<xsd:documentation>
Name of the variable.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="DisplayName" type="StringT64">
<xsd:annotation>
<xsd:documentation>
Name of the variable used for display on UI.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
<xsd:attribute name="required" type="xsd:boolean" default="false">
<xsd:annotation>
<xsd:documentation>
A Flag indicating whether or not the variable is mandatory.
</xsd:documentation>
</xsd:annotation>
</xsd:attribute>
</xsd:complexType>
```

```
<xsd:complexType name="ExternalURLType">
<xsd:annotation>
<xsd:documentation>
This section defines a complex type for external URL.
</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element name="Pattern" type="StringT256">
<xsd:annotation>
<xsd:documentation>
The URL pattern used to format links to the external server.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="ConfigVariable" type="ConfigVariableType" minOccurs="0"
maxOccurs="50">
<xsd:annotation>
<xsd:documentation>
An optional list of configuration variablees representing the details
of the external server. They are used for constructing links to
the server based on the URL pattern.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="TemplateRegistrationType">
<xsd:annotation>
<xsd:documentation>
This section defines a complex type for template registration metadata
which is used to register templates during connector deployment.
</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element name="FileName" type="StringT256">
<xsd:annotation>
<xsd:documentation>
The template file name.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="InternalName" type="StringStrictT128">
<xsd:annotation>
<xsd:documentation>
A name representing the template in the connector framework.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="TemplateName" type="StringStrictT128">
<xsd:annotation>
<xsd:documentation>
The template display name to be used on UI.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="TemplateType">
<xsd:annotation>
<xsd:documentation>
The template type as one of the three options defined next.
</xsd:documentation>
```

```
</xsd:annotation>
<xsd:simpleType>
<xsd:restriction base="xsd:string">
<xsd:enumeration value="InboundXSL"/>
<xsd:enumeration value="OutboundXSL"/>
<xsd:enumeration value="OutboundXML"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="Description" type="StringT512">
<xsd:annotation>
<xsd:documentation>
A description of the template.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

Example A–9 EMIncident.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://xmlns.oracle.com/sysman/connector"
targetNamespace="http://xmlns.oracle.com/sysman/connector"
elementFormDefault="qualified">
<xsd:include schemaLocation="../common/connectorCommon.xsd"/>
<xsd:element name="EMIncident">
<xsd:annotation>
<xsd:documentation>
This section provides a data structure based on EM incidents for all
ticketing actions.
</xsd:documentation>
</xsd:annotation>
<xsd:complexType>
<xsd:sequence>
<xsd:element name="ConnectorGUID" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
A unique ID to identify the connector that is processing
the incident.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="TicketID" type="xsd:string" minOccurs="0">
<xsd:annotation>
<xsd:documentation>
The ID to identify the ticket created in the external ticketing system.
It is generated in the external system and used to update
the ticket.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="HDUser" type="xsd:string" minOccurs="0">
<xsd:annotation>
<xsd:documentation>
HelpDesk user name provided from UI during connector configuration.
</xsd:documentation>
</xsd:annotation>
```

```
</xsd:element>
<xsd:element name="NotificationRuleOwner" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
The owner of the notification rule which generated the incident.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="NotificationRuleName" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
The name of the notification rule which generated the incident.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="ReopenTicket" type="xsd:string" minOccurs="0">
<xsd:annotation>
<xsd:documentation>
The identifier of the previous ticket that should be re-opened.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="ConnectorVariable" type="VariableType"
minOccurs="0" maxOccurs="50">
<xsd:annotation>
<xsd:documentation>
An optinal list of up ro 50 connector variables that contain
name/value pairs.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Property" type="PropertyType" minOccurs="0"
maxOccurs="50">
<xsd:annotation>
<xsd:documentation>
An optinal list of up ro 50 property variables as defined in
connectorCommon.xsd.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="SystemAttributes"
type="IncidentSystemAttributesType">
<xsd:annotation>
<xsd:documentation>
A list of attributes for incidents as defined by EM event system.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:complexType name="IncidentSystemAttributesType">
<xsd:annotation>
<xsd:documentation>
This section defines a complex type for incident attributes provided by
EM event system.
</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element name="IncidentID" type="xsd:string">
```

```
<xsd:annotation>
<xsd:documentation>
The ID of an incident.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="SourceInfo" type="SourceInfoType"
maxOccurs="unbounded">
<xsd:annotation>
<xsd:documentation>
The source information of the EM subsystems or componenets that
raises the incident.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="IncidentURL" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
A URL to the incident on EM.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="AutoClose" type="xsd:boolean">
<xsd:annotation>
<xsd:documentation>
A flag indicating if an incident is auto closed by the system, or it
has to be manually closed by users.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="TicketStatus" type="xsd:string" minOccurs="0">
<xsd:annotation>
<xsd:documentation>
If an external ticket is associated with the incident,
the status of the ticket as assigned at an external help desk system,
and updated in EM.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Owner" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
User to whom the incident is assigned to resolve the issue.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="ResolutionState" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
The attribute used to track where the incident is in terms of resolution.
For instance, it can be "new" or "closed".
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Acknowledge" type="xsd:boolean">
<xsd:annotation>
<xsd:documentation>
A flag indicating whether or not the incident has been acknowledged.
Acknowledgement is simply a way for an administrator to indicate
that they have viewed the incident and take ownership of it.
```

```
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Escalated" type="xsd:boolean">
<xsd:annotation>
<xsd:documentation>
A flag indicating whether or not the incident has been escalated.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="EscalationLevel" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
The hierarchical level of escalation that has been made to this incident.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Priority" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
The priority order in which the issue should be resolved.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Summary" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
A text summary of the incident.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="CreationDate" type="xsd:dateTime">
<xsd:annotation>
<xsd:documentation>
The time when the incident is created by associating event to incident.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="LastUpdatedDate" type="xsd:dateTime">
<xsd:annotation>
<xsd:documentation>
The time when the incident is last updated. The incident update
includes changes to any of the tracking attributes or changes to
the associated events and event sequence.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Category" type="xsd:string" minOccurs="0"
maxOccurs="50">
<xsd:annotation>
<xsd:documentation>
An optional list of categories of the incidents.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="LastModifiedBy" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
User who last modified the incident.
</xsd:documentation>
```

```
</xsd:annotation>
</xsd:element>
<xsd:element name="Severity" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
Severity level of the incident. The value changes based on local language setting.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="SeverityCode" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
Internal Severity value of the current event.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

Example A–10 connectorCommon.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://xmlns.oracle.com/sysman/connector"
targetNamespace="http://xmlns.oracle.com/sysman/connector"
elementFormDefault="qualified">
<xsd:include schemaLocation="externalEvent.xsd"/>
<xsd:complexType name="SourceInfoType">
<xsd:annotation>
<xsd:documentation>
This section defines a complext type for Source Information.
</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element name="SourceObjInfo" type="SourceObjInfoType" minOccurs="0">
<xsd:annotation>
<xsd:documentation>
This element defines the data structure for the source object, the EM
subsystem or component, that raises an EM event or an incident.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="TargetInfo" type="TargetInfoType" minOccurs="0">
<xsd:annotation>
<xsd:documentation>
The element defines the data structure for an EM target as related
to the connector framework.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SourceObjInfoType">
<xsd:annotation>
<xsd:documentation>
This section defines a complex type for Source Object Information.
</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
```

```
<xsd:element name="ObjID" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
The unique ID to identify the source object.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="ObjName" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
The name of the source object.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="ObjOwner" type="xsd:string" minOccurs="0">
<xsd:annotation>
<xsd:documentation>
The owner of the source object.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="SourceObjType" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
The type of the source object.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="SourceObjSubType" type="xsd:string" minOccurs="0">
<xsd:annotation>
<xsd:documentation>
The subtype of the source object.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="TargetInfoType">
<xsd:annotation>
<xsd:documentation>
This section defines a complex type for target information.
</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element name="TargetGUID" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
A unique GUID for the target.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="TargetName" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
Name of the target.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="TargetType" type="xsd:string">
<xsd:annotation>
```

```
<xsd:documentation>
Type of the target.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="TargetTypeLabel" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
The display label of the target type.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="TargetURL" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
The URL of the target.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="TargetProperty" type="PropertyType" minOccurs="0"
maxOccurs="50">
<xsd:annotation>
<xsd:documentation>
An optional list of properties for the target.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="PropertyType">
<xsd:annotation>
<xsd:documentation>
This section defines a complex type for a property attribute.
</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element name="Name" type="xsd:string">
<xsd:annotation>
<xsd:documentation>
A string name defining a property attribute.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="Value" type="xsd:string" nillable="true">
<xsd:annotation>
<xsd:documentation>
A non-null string value.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="VariableType">
<xsd:annotation>
<xsd:documentation>
This section defines a complex type for a general variable.
</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element name="VariableName" type="StringStrictT32">
```

```
<xsd:annotation>
<xsd:documentation>
Name of the variable. It has to be a string containing 1 or upto
32 upper case or lower case letters or numbers.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="VariableValue" type="StringT2048">
<xsd:annotation>
<xsd:documentation>
Value of the variable. It has to be a string containing 1 or upto
2048 characters.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="GetAlertsResponse">
<xsd:annotation>
<xsd:documentation>
This section defines a complex type for responses to a getAlerts request.
</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element name="Alert" minOccurs="0" maxOccurs="200">
<xsd:annotation>
<xsd:documentation>
The individual alerts contained in the response. A response may have
upto 200 alerts.
</xsd:documentation>
</xsd:annotation>
<xsd:complexType>
<xsd:sequence>
<xsd:element ref="ExternalEvent">
<xsd:annotation>
<xsd:documentation>
Details of the external event in the alert, as defined in
ExternalEvent.xsd.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="InstanceVariable" type="VariableType"
minOccurs="0" maxOccurs="50">
<xsd:annotation>
<xsd:documentation>
A list of instance variables for the alert.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ConnectorVariablesType">
<xsd:annotation>
<xsd:documentation>
This section defines a complex type for connector variables. An element
of type ConnectorVariablesType may have up to 50 connector variables, as
defined next.
```

```
</xsd:documentation>
</xsd:annotation>
<xsd:sequence>
<xsd:element name="ConnectorVariable" type="VariableType" minOccurs="0"
maxOccurs="50">
<xsd:annotation>
<xsd:documentation>
A connector varialbe as a name/value pair.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="StringT64">
<xsd:annotation>
<xsd:documentation>
This section defines a simple type for a String with maximum length of
64 bytes.
</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
<xsd:minLength value="1"/>
<xsd:maxLength value="64"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringT128">
<xsd:annotation>
<xsd:documentation>
This section defines a simple type for a String with maximum length of
128 bytes.
</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
<xsd:minLength value="1"/>
<xsd:maxLength value="128"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringT256">
<xsd:annotation>
<xsd:documentation>
This section defines a simple type for a String with maximum length of
256 bytes.
</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
<xsd:minLength value="1"/>
<xsd:maxLength value="256"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringT512">
<xsd:annotation>
<xsd:documentation>
This section defines a simple type for a String with maximum length of
512 bytes.
</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
<xsd:minLength value="1"/>
<xsd:maxLength value="512"/>
</xsd:restriction>
```

```
</xsd:simpleType>
<xsd:simpleType name="StringT2048">
<xsd:annotation>
<xsd:documentation>
This section defines a simple type for a String with maximum length of
2048 bytes.
</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
<xsd:minLength value="1"/>
<xsd:maxLength value="2048"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringStrictT16">
<xsd:annotation>
<xsd:documentation>
This section defines a simple type for a String with maximum length of
16 bytes. The String can only contain lower or upper case letters, numbers,
and the underscore characters.
</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
<xsd:minLength value="1"/>
<xsd:maxLength value="16"/>
<xsd:pattern value="([a-zA-Z0-9_])*"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringStrictT32">
<xsd:annotation>
<xsd:documentation>
This section defines a simple type for a String with maximum length of
32 bytes. The String can only contain lower or upper case letters, numbers,
and the underscore characters.
</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
<xsd:minLength value="1"/>
<xsd:maxLength value="32"/>
<xsd:pattern value="([a-zA-Z0-9_])*"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringStrictT64">
<xsd:annotation>
<xsd:documentation>
This section defines a simple type for a String with maximum length of
64 bytes. The String can only contain lower or upper case letters, numbers,
and the underscore characters.
</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
<xsd:minLength value="1"/>
<xsd:maxLength value="64"/>
<xsd:pattern value="([a-zA-Z0-9_])*"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringStrictT128">
<xsd:annotation>
<xsd:documentation>
This section defines a simple type for a String with maximum length of
128 bytes. The String can only contain lower or upper case letters, numbers,
and the underscore characters.
</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
<xsd:minLength value="1"/>
<xsd:maxLength value="128"/>
<xsd:pattern value="([a-zA-Z0-9_])*"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="StringStrictT256">
<xsd:annotation>
<xsd:documentation>
This section defines a simple type for a String with maximum length of
256 bytes. The String can only contain lower or upper case letters, numbers,
and the underscore characters.
</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
<xsd:minLength value="1"/>
<xsd:maxLength value="256"/>
<xsd:pattern value="([a-zA-Z0-9_])*"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="VersionT">
<xsd:annotation>
<xsd:documentation>
This section defines a simple type for a String with maximum length of
20 bytes. The String can only contain numbers and the period characters.
</xsd:documentation>
</xsd:annotation>
<xsd:restriction base="xsd:string">
<xsd:minLength value="1"/>
<xsd:maxLength value="20"/>
<xsd:pattern value="([0-9.])*"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:schema>
```

Example A–11 createTicket_response.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://xmlns.oracle.com/sysman/connector"
targetNamespace="http://xmlns.oracle.com/sysman/connector"
elementFormDefault="qualified">
<xsd:include schemaLocation="connectorCommon.xsd"/>
<xsd:element name="CreateTicketResponse">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="ticketID" type="StringT128"/>
<xsd:element name="InstanceVariable" type="VariableType" minOccurs="0"
maxOccurs="50"/>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

Example A–12 getTicket_response.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://xmlns.oracle.com/sysman/connector"
targetNamespace="http://xmlns.oracle.com/sysman/connector"
elementFormDefault="qualified">
<xsd:include schemaLocation="../common/connectorCommon.xsd"/>
<xsd:element name="getTicketResponse">
<xsd:annotation>
<xsd:documentation>
The response from the external system upon a getTicketRequest request.
It must contain a ticket ID from the output of the request Web Service.
</xsd:documentation>
</xsd:annotation>
<xsd:complexType>
<xsd:sequence>
<xsd:element name="ticketID" type="StringT128">
<xsd:annotation>
<xsd:documentation>
The ticket ID.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

Example A–13 publishTicket.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://xmlns.oracle.com/sysman/connector"
targetNamespace="http://xmlns.oracle.com/sysman/connector"
elementFormDefault="qualified">
<xsd:include schemaLocation="../common/connectorCommon.xsd"/>
<xsd:element name="publishTicketStatus">
<xsd:annotation>
<xsd:documentation>
This section defines the request to publish ticket status from Ticketing
system to EM when it is updated.
</xsd:documentation>
</xsd:annotation>
<xsd:complexType>
<xsd:sequence>
<xsd:element name="ConnectorGUID" type="StringStrictT16">
<xsd:annotation>
<xsd:documentation>
The GUID of the connector the request is to be sent to. The GUID
is communicated to the external system in the earlier requests to
create tickets. It is returned in the inbound data to associate
the date with the corresponding ticket.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="ticketID" type="StringT128">
<xsd:annotation>
<xsd:documentation>
The ID of the ticket whose status is being updated.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
```

```
<xsd:element name="ticketStatus" type="StringT64">
<xsd:annotation>
<xsd:documentation>
The new status of the ticket.
</xsd:documentation>
</xsd:annotation>
</xsd:element>
<xsd:element name="RequestOccurTS" type="StringT64">
<xsd:annotation>
<xsd:documentation>
Time when the inbound call is invoked.
</xsd:documentation>
</xsd:annotation>
</xsd:element> <!-- NOT used xsd:dateTime because anyway sql date format must be
used -->
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

B

Error Messages and Debugging

This chapter provides all Management Connector specific error messages and debugging information. The errors are returned in the response model.

Error Messages

This section provides error codes, descriptions, causes, and suggested actions for all Connector Framework error messages.

CNTR-0001

Cause: Authentication failed. The credential to log in to Enterprise Manager is incorrect.

Action: Correct the Enterprise Manager credential element in the request.

CNTR-0002

Cause: In `setModel`, the requested aggregate target list is of size 0. This operation is not supported. In `setModel`, the requested aggregate target list is of size 0. This operation is not supported.

Action: Correct the request model to include two aggregate targets: one of type cluster, and the other of type `rac_database`.

CNTR-0003

Cause: The current cluster aggregate target and `rac_database` aggregate target have a different number of member targets.

Action: Make sure the numbers of members of the current cluster aggregate target and the `rac_database` aggregate target are the same.

CNTR-0004

Cause: The target name or type element is NULL in the request model for `getModel`.

Action: Correct the target name or target type in the request model.

CNTR-0005

Cause: There is an unrecognized property in the request model for `getModel`.

Action: Remove the unrecognized property.

CNTR-0006

Cause: The name or type of element of an aggregate target is NULL.

Action: Correct the name or type of the aggregate target in the request model.

CNTR-0007

Cause: The aggregate target type is something other than `cluster` and `rac_database`.

Action: Correct the aggregate target type. Only two types are supported in this release: `cluster` and `rac_database`.

CNTR-0008

Cause: The request model is invalid.

Action: Correct the request model. Make sure the request model has either zero or two aggregate targets (one of type `cluster` and one of type `rac_database`). If aggregate targets are included, make sure the numbers of targets in the two aggregate targets are the same with the same set of hosts. Make sure the name and host elements of each target in the cluster aggregate target are the same.

CNTR-0009

Cause: Either the `cluster` aggregate target or the `rac_database` aggregate target is missing in the request model.

Action: Add the missing aggregate target.

CNTR-0010

Cause: No software library image was found based on the description of the model.

Action: Correct the name of the software library image, or make sure the image is available in the Enterprise Manager software library.

CNTR-0011

Cause: No existing node could be found to run some steps of the add node job.

Action: Correct the request model to make sure all existing nodes are specified correctly in the request model.

CNTR-0012

Cause: No credential was specified for the RAC nodes.

Action: Add the credentials for the RAC nodes.

CNTR-0013

Cause: The name of the member target of the rac_database aggregate target does not follow the `<rac_name>_<instance_name>` naming rule.

Action: Correct the name of the member target of the `rac_database` aggregate target.

CNTR-0014

Cause: The storage element was missing during the RAC creation request.

Action: Add the storage element in the request model.

CNTR-0015

Cause: The number of nodes in the request model is less than the number of nodes in the current model minus one.

Action: Correct the response model by deleting only one node.

CNTR-0016

Cause: The `cluster` aggregate target and `rac_database` aggregate target have a different number of member targets.

Action: Correct the request model with the correct member targets for both the `cluster` aggregate target and `rac_database` aggregate target.

CNTR-0017

Cause: More than one node was specified in the request when the RAC database did not exist yet.

Action: Correct the request model to use only one node for the new RAC database.

CNTR-0018

Cause: The member targets of the aggregate targets do not match those in the Enterprise Manager repository.

Action: Correct the request model with the correct member target names.

CNTR-0019

Cause: Nothing to do: there are no member differences between the current and requested model. The members of the current model inside Enterprise Manager are the same as the one in the request. The connector cannot infer any action.

Action: Correct the request model to indicate a provisioning action.

CNTR-0020

Cause: The number of nodes in the request model is more than the number of nodes in the current model plus one.

Action: Correct the request model by adding only one node.

CNTR-0022

Cause: There is an error in the host of RAC aggregate target of the request model. The host attribute of the member targets does not match the host attribute in the Enterprise Manager repository.

Action: Correct the host attribute of the member target of the `rac_database` aggregate target.

CNTR-0023

Cause: There is an error in the host of the Oracle Clusterware aggregate target of the request model. The host attribute of the member targets does not match the host attribute in the Enterprise Manager repository.

Action: Correct the host attribute of the member target of the `cluster` aggregate target.

CNTR-0024

Cause: The host attribute of the member target of the `cluster` aggregate target does not match the host attribute for the corresponding member target of the `rac_database` aggregate target.

Action: Correct the host attribute of the member target of the `cluster` and `rac_database` aggregate targets.

CNTR-0025 (Windows only)

Cause: The `sl_OHPartitionsAndSpace_valueFromDlg property` is missing from the cluster aggregate target properties.

Action: Add the `sl_OHPartitionsAndSpace_valueFromDlg` property to the cluster aggregate target properties in the request model.

CNTR-0026 (Windows only)

Cause: The `ret_PrivIntrList` property is missing from the cluster aggregate target properties.

Action: Add the `ret_PrivIntrList` property to the cluster aggregate target properties in the request model.

Debugging

The Connector Framework uses the log4j logging utility to log the types of messages shown in Table B–1:

Table B–1 Message Types and Corresponding Code Names

Message Type	Code Option
Warning	WARN
Error	ERROR
Debugging	DEBUG
Information	INFO

Specifying the Debug Option

The following example shows the insertion of DEBUG in the following file:

```
$ORACLE_HOME/sysman/config/emomslogging.properties
```

Use the following emctl command to set the debug level as shown below:

```
emctl set property -name log4j.rootCategory -value "DEBUG,
emlogAppender, emtrcAppender" -module emoms
```

Enter the SYSMAN password when prompted.

Viewing Debug Messages

The debug messages from the Connector Framework are displayed in the following file:

```
  $INSTANCE_HOME/sysman/log/emoms.trc
```

C

MOM Event Connector Samples

This appendix provides sample implementations for a MOM event connector.

Example C–1 connectorDeploy.xml

```
<?xml version="1.0" encoding="UTF-8" ?>

<ManagementConnector xmlns="http://xmlns.oracle.com/sysman/connector">
  <Name>Microsoft Operations Manager Connector</Name>
  <Version>12.1.0.1.0</Version>
  <EMCompatibleVersion>12.1.0.1.0</EMCompatibleVersion>
  <Description>Microsoft Operations Manager Integration with Enterprise
Manager</Description>
  <Category>EventConnector</Category>
  <ConfigVariable required="true">
    <VariableName>SETUP_NAME</VariableName>
    <DisplayName>MOM Registered Connector Name</DisplayName>
  </ConfigVariable>
  <ConfigVariable required="true">
    <VariableName>RESOLUTION_STATE</VariableName>
    <DisplayName>Resolution State</DisplayName>
  </ConfigVariable>
  <Service>
    <Method>setup</Method>
    <WebServiceEndpoint>
      <![CDATA[http://[server]:1271/ConnectorServiceV2.asmx]]>
    </WebServiceEndpoint>

<SOAPAction>http://www.microsoft.com/EnterpriseManagement/Mom/Connector/V2/SetupWi
thResolutionState</SOAPAction>
  </Service>
  <Service>
    <Method>initialize</Method>
    <WebServiceEndpoint>
      <![CDATA[http://[server]:1271/ConnectorServiceV2.asmx]]>
    </WebServiceEndpoint>

<SOAPAction>http://www.microsoft.com/EnterpriseManagement/Mom/Connector/V2/Initial
ize</SOAPAction>
  </Service>
  <Service>
    <Method>createEvent</Method>
    <WebServiceEndpoint>
      <![CDATA[http://[server]:1271/ConnectorServiceV2.asmx]]>
    </WebServiceEndpoint>

<SOAPAction>http://www.microsoft.com/EnterpriseManagement/Mom/Connector/V2/InsertA
```

```
lerts</SOAPAction>
  </Service>
  <Service>
    <Method>updateEvent</Method>
    <WebServiceEndpoint>
      <![CDATA[http://[server]:1271/ConnectorServiceV2.asmx]]>
    </WebServiceEndpoint>

<SOAPAction>http://www.microsoft.com/EnterpriseManagement/Mom/Connector/V2/UpdateA
lerts</SOAPAction>
  </Service>
  <Service>
    <Method>uninitialize</Method>
    <WebServiceEndpoint>
      <![CDATA[http://[server]:1271/ConnectorServiceV2.asmx]]>
    </WebServiceEndpoint>

<SOAPAction>http://www.microsoft.com/EnterpriseManagement/Mom/Connector/V2/Uniniti
alize</SOAPAction>
  </Service>
  <Service>
    <Method>cleanup</Method>
    <WebServiceEndpoint>
      <![CDATA[http://[server]:1271/ConnectorServiceV2.asmx]]>
    </WebServiceEndpoint>

<SOAPAction>http://www.microsoft.com/EnterpriseManagement/Mom/Connector/V2/Cleanup
</SOAPAction>
  </Service>
  <TemplateRegistration>
    <FileName>setup_request.xml</FileName>
    <InternalName>setup</InternalName>
    <TemplateName>Setup</TemplateName>
    <TemplateType>OutboundXML</TemplateType>
    <Description>This is the request xml file for setup method</Description>
  </TemplateRegistration>
  <TemplateRegistration>
    <FileName>setup_response.xsl</FileName>
    <InternalName>setup</InternalName>
    <TemplateName>Setup</TemplateName>
    <TemplateType>InboundXSL</TemplateType>
    <Description>This is the response xsl file for setup method</Description>
  </TemplateRegistration>
  <TemplateRegistration>
    <FileName>initialize_request.xml</FileName>
    <InternalName>initialize</InternalName>
    <TemplateName>Initialize</TemplateName>
    <TemplateType>OutboundXML</TemplateType>
   <Description>This is the request xml file for initialize method</Description>
  </TemplateRegistration>
  <TemplateRegistration>
    <FileName>createEvent_request.xsl</FileName>
    <InternalName>createEvent</InternalName>
    <TemplateName>Create Event</TemplateName>
    <TemplateType>OutboundXSL</TemplateType>
   <Description>This is the request xsl file for createEvent method</Description>
  </TemplateRegistration>
  <TemplateRegistration>
    <FileName>createEvent_response.xsl</FileName>
    <InternalName>createEvent</InternalName>
    <TemplateName>Create Event</TemplateName>
    <TemplateType>InboundXSL</TemplateType>
    <Description>This is the response xsl file for createEvent
method</Description>
  </TemplateRegistration>
  <TemplateRegistration>
    <FileName>updateEvent_request.xsl</FileName>
    <InternalName>updateEvent</InternalName>
    <TemplateName>Update Event</TemplateName>
    <TemplateType>OutboundXSL</TemplateType>
   <Description>This is the request xsl file for updateEvent method</Description>
  </TemplateRegistration>
  <TemplateRegistration>
    <FileName>updateEvent_response.xsl</FileName>
    <InternalName>updateEvent</InternalName>
    <TemplateName>Update Event</TemplateName>
    <TemplateType>InboundXSL</TemplateType>
    <Description>This is the response xsl file for updateEvent
method</Description>
  </TemplateRegistration>
  <TemplateRegistration>
    <FileName>uninitialize_request.xml</FileName>
    <InternalName>uninitialize</InternalName>
    <TemplateName>Uninitializ</TemplateName>
    <TemplateType>OutboundXML</TemplateType>
    <Description>This is the request xml file for uninitialize
method</Description>
  </TemplateRegistration>
  <TemplateRegistration>
    <FileName>cleanup_request.xml</FileName>
    <InternalName>cleanup</InternalName>
    <TemplateName>Cleanup</TemplateName>
    <TemplateType>OutboundXML</TemplateType>
    <Description>This is the request xml file for cleanup method</Description>
  </TemplateRegistration>
</ManagementConnector>
```

Example C–2 setup_request.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<SetupWithResolutionState
xmlns="http://www.microsoft.com/EnterpriseManagement/Mom/Connector/V2">
  <forwardeeInfo>
    <ForwardeeName>$SETUP_NAME$</ForwardeeName>
  </forwardeeInfo>
  <resolutionState>$RESOLUTION_STATE$</resolutionState>
</SetupWithResolutionState>
```

Example C–3 setup_response.xsl

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:transform version="1.0"
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns:a="http://www.microsoft.com/EnterpriseManagement/Mom/Connector/V2"
                xmlns:ns0="http://xmlns.oracle.com/sysman/connector">
 <xsl:template match="/">
  <ns0:SetupResponse>
   <ns0:ConnectorVariable>
    <ns0:VariableName>REGISTRATION_ID</ns0:VariableName>
    <ns0:VariableValue>
```

```
     <xsl:value-of
select="a:SetupWithResolutionStateResponse/a:SetupWithResolutionStateResult"/>
    </ns0:VariableValue>
   </ns0:ConnectorVariable>
  </ns0:SetupResponse>
 </xsl:template>
</xsl:transform>
```

Example C–4 initialize_request.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<Initialize
xmlns="http://www.microsoft.com/EnterpriseManagement/Mom/Connector/V2">
  <registrationId>$REGISTRATION_ID$</registrationId>
  <dataChangeFlags>NewAlerts UpdatedAlerts</dataChangeFlags>
</Initialize>
```

Example C–5 uninitialize_request.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<Uninitialize
xmlns="http://www.microsoft.com/EnterpriseManagement/Mom/Connector/V2">
  <registrationId>$REGISTRATION_ID$</registrationId>
  <dataChangeFlags>NewAlerts UpdatedAlerts</dataChangeFlags>
</Uninitialize>
```

Example C–6 cleanup_request.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<Cleanup xmlns="http://www.microsoft.com/EnterpriseManagement/Mom/Connector/V2">
  <registrationId>$REGISTRATION_ID$</registrationId>
</Cleanup>
```

Example C–7 createEvent_request.xsl

```
<?xml version='1.0' ?>
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns:ns0="http://www.microsoft.com/EnterpriseManagement/Mom/Connector/V2"
               xmlns:a="http://xmlns.oracle.com/sysman/connector">
    <xsl:template match="a:EMEvent">
        <ns0:InsertAlerts>
            <ns0:registrationId>
                <xsl:for-each select="a:ConnectorVariable">
                    <xsl:if test="a:VariableName = 'REGISTRATION_
ID'"><xsl:value-of select="a:VariableValue"/></xsl:if>
                </xsl:for-each>
             </ns0:registrationId>
            <ns0:alerts>
                <ns0:AlertInsert>
                    <ns0:Name><xsl:value-of
select="a:SystemAttributes/a:EventClass"/></ns0:Name>
                    <ns0:ComputerName>
                      <xsl:choose>
<xsl:when
test="normalize-space(a:SystemAttributes/a:SourceInfo/a:TargetInfo/a:TargetName)
!= ''">
                                <xsl:value-of
select="a:SystemAttributes/a:SourceInfo/a:TargetInfo/a:TargetName"/>
                          </xsl:when>
                          <xsl:otherwise>N/A</xsl:otherwise>
                      </xsl:choose>
                    </ns0:ComputerName>
                    <ns0:ComputerDomain></ns0:ComputerDomain>
                    <ns0:Description>
                        Received event reported by Oracle Enterprise Manager:

                        Occurred Date: <xsl:choose><xsl:when
test="normalize-space(a:SystemAttributes/a:OccurredDate) != ''"><xsl:value-of
select="a:SystemAttributes/a:OccurredDate"/></xsl:when><xsl:otherwise>N/A</xsl:oth
erwise></xsl:choose>
                        Reported Date: <xsl:choose><xsl:when
test="normalize-space(a:SystemAttributes/a:ReportedDate) != ''"><xsl:value-of
select="a:SystemAttributes/a:ReportedDate"/></xsl:when><xsl:otherwise>N/A</xsl:oth
erwise></xsl:choose>
                        Event Class: <xsl:value-of
select="a:SystemAttributes/a:EventClass"/>
                        Event Name: <xsl:value-of
select="a:SystemAttributes/a:EventName"/>
                        Target Type: <xsl:value-of
select="a:SystemAttributes/a:SourceInfo/a:TargetInfo/a:TargetType"/>
                        Target Name: <xsl:value-of
select="a:SystemAttributes/a:SourceInfo/a:TargetInfo/a:TargetName"/>
                        Severity: <xsl:value-of
select="a:SystemAttributes/a:Severity"/>
                        Message: <xsl:value-of
select="a:SystemAttributes/a:Message"/>
                        URL: <xsl:value-of
select="a:SystemAttributes/a:EventURL"/>
                        Event Context:
                        <xsl:for-each select="a:EventContextAttributes">
                            <xsl:choose>
                                <xsl:when test="a:StringAttribute"><xsl:value-of
select="a:StringAttribute/a:Name"/>: <xsl:value-of
select="a:StringAttribute/a:Value"/></xsl:when>
                                <xsl:when test="a:NumberAttribute"><xsl:value-of
select="a:NumberAttribute/a:Name"/>: <xsl:value-of
select="a:NumberAttribute/a:Value"/></xsl:when>
                            </xsl:choose>
                        </xsl:for-each>
                    </ns0:Description>
                    <ns0:Severity>
                        <xsl:choose>
                            <xsl:when test="a:SystemAttributes/a:SeverityCode =
'CLEAR'">Success</xsl:when>
                            <xsl:when test="a:SystemAttributes/a:SeverityCode =
'INFORMATIONAL'">Information</xsl:when>
                            <xsl:when test="a:SystemAttributes/a:SeverityCode =
'WARNING'">Warning</xsl:when>
                            <xsl:when test="a:SystemAttributes/a:SeverityCode =
'MINOR_WARNING'">Warning</xsl:when>
                            <xsl:when test="a:SystemAttributes/a:SeverityCode =
'CRITICAL'">CriticalError</xsl:when>
                            <xsl:when test="a:SystemAttributes/a:SeverityCode =
'FATAL'">CriticalError</xsl:when>
                            <xsl:otherwise>Error</xsl:otherwise>
                        </xsl:choose>
                    </ns0:Severity>
                    <CustomField1></CustomField1>
                    <CustomField2></CustomField2>
                    <CustomField3></CustomField3>
                    <CustomField4></CustomField4>
                    <CustomField5></CustomField5>
                    <ns0:RuleId>00000000-0000-0000-0000-000000000000</ns0:RuleId>
                    <ns0:Source>
                        <xsl:value-of
select="a:SystemAttributes/a:SourceInfo/a:TargetInfo/a:TargetType"/>:
<xsl:value-of select="a:SystemAttributes/a:SourceInfo/a:TargetInfo/a:TargetName"/>
                    </ns0:Source>
                    <ns0:TimeRaised>
                      <xsl:choose>
                        <xsl:when
test="normalize-space(a:SystemAttributes/a:OccurredDate) != ''">
                          <xsl:value-of
select="a:SystemAttributes/a:OccurredDate"/>
                        </xsl:when>
                        <xsl:when
test="normalize-space(a:SystemAttributes/a:ReportedDate) != ''">
                          <xsl:value-of
select="a:SystemAttributes/a:ReportedDate"/>
                        </xsl:when>
                        <xsl:otherwise>N/A</xsl:otherwise>
                      </xsl:choose>
                    </ns0:TimeRaised>
                    <ns0:ResolutionState>0</ns0:ResolutionState>
                    <ns0:ServerRole></ns0:ServerRole>
                    <ns0:ServerRoleInstance></ns0:ServerRoleInstance>
                    <ns0:Component></ns0:Component>
                  <ns0:ComponentInstance></ns0:ComponentInstance>
                </ns0:AlertInsert>
            </ns0:alerts>
        </ns0:InsertAlerts>
    </xsl:template>
</xsl:transform>
```

Example C–8   createEvent_response.xsl

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns:a="http://www.microsoft.com/EnterpriseManagement/Mom/Connector/V2"
               xmlns:ns0="http://xmlns.oracle.com/sysman/connector">
<xsl:template match="a:InsertAlertsResponse/a:InsertAlertsResult">
  <ns0:EMEventResponse>
   <xsl:choose>
    <xsl:when test="a:InsertedAlerts/a:SuccessfulAlertInsert/a:NewAlertId">
     <ns0:SuccessFlag>true</ns0:SuccessFlag>
     <ns0:ExternalEventId>
      <xsl:value-of
select="a:InsertedAlerts/a:SuccessfulAlertInsert/a:NewAlertId"/>
     </ns0:ExternalEventId>
    </xsl:when>
    <xsl:otherwise>
     <ns0:SuccessFlag>false</ns0:SuccessFlag>
     <ns0:ErrorMessage>
      <xsl:value-of select="a:FailedAlerts/a:FailedAlertInsert/a:Error"/>
     </ns0:ErrorMessage>
    </xsl:otherwise>
   </xsl:choose>
  </ns0:EMEventResponse>
 </xsl:template>
```

```
</xsl:transform>
```

Example C–9 updateEvent_request.xsl

```
<?xml version='1.0' ?>
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns:ns0="http://www.microsoft.com/EnterpriseManagement/Mom/Connector/V2"
               xmlns:a="http://xmlns.oracle.com/sysman/connector">
    <xsl:template match="a:EMEvent">
        <ns0:UpdateAlerts>
            <ns0:registrationId>
                <xsl:for-each select="a:ConnectorVariable">
                    <xsl:if test="a:VariableName = 'REGISTRATION_
ID'"><xsl:value-of select="a:VariableValue"/></xsl:if>
                </xsl:for-each>
            </ns0:registrationId>
            <ns0:updatedAlerts>
                <ns0:AlertUpdate>
                    <ns0:AlertId><xsl:value-of
select="a:ExternalEventID"/></ns0:AlertId>
                    <ns0:OwnerNameUseExisting>true</ns0:OwnerNameUseExisting>
                    <ns0:SeverityUseExisting>false</ns0:SeverityUseExisting>
                    <xsl:variable name="MOMSeverity">
                        <xsl:choose>
                            <xsl:when test="a:SystemAttributes/a:SeverityCode =
'CLEAR'">Success</xsl:when>
                            <xsl:when test="a:SystemAttributes/a:SeverityCode =
'INFORMATIONAL'">Information</xsl:when>
                            <xsl:when test="a:SystemAttributes/a:SeverityCode =
'WARNING'">Warning</xsl:when>
                            <xsl:when test="a:SystemAttributes/a:SeverityCode =
'MINOR_WARNING'">Warning</xsl:when>
                            <xsl:when test="a:SystemAttributes/a:SeverityCode =
'CRITICAL'">CriticalError</xsl:when>
                            <xsl:when test="a:SystemAttributes/a:SeverityCode =
'FATAL'">CriticalError</xsl:when>
                            <xsl:otherwise>Error</xsl:otherwise>
                        </xsl:choose>
                    </xsl:variable>
                    <ns0:Severity>
                        <xsl:value-of select="$MOMSeverity"/>
                    </ns0:Severity>
                    <xsl:choose>
                        <xsl:when test="$MOMSeverity = 'Success'">

<ns0:ResolutionStateUseExisting>false</ns0:ResolutionStateUseExisting>
                            <ns0:ResolutionState>255</ns0:ResolutionState>
                        </xsl:when>
                        <xsl:otherwise>

<ns0:ResolutionStateUseExisting>true</ns0:ResolutionStateUseExisting>
                            <ns0:ResolutionState>0</ns0:ResolutionState>
                        </xsl:otherwise>
                    </xsl:choose>

<ns0:CustomField1UseExisting>true</ns0:CustomField1UseExisting>

<ns0:CustomField2UseExisting>true</ns0:CustomField2UseExisting>
```

```
<ns0:CustomField3UseExisting>true</ns0:CustomField3UseExisting>

<ns0:CustomField4UseExisting>true</ns0:CustomField4UseExisting>

<ns0:CustomField5UseExisting>true</ns0:CustomField5UseExisting>

<ns0:LastModifiedByUseExisting>true</ns0:LastModifiedByUseExisting>
                    <ns0:RepeatCountUseExisting>true</ns0:RepeatCountUseExisting>
                    <ns0:RepeatCount>0</ns0:RepeatCount>
                    <ns0:TicketIDUseExisting>true</ns0:TicketIDUseExisting>

<ns0:TimeOfFirstEventUseExisting>true</ns0:TimeOfFirstEventUseExisting>

<ns0:TimeOfLastEventUseExisting>true</ns0:TimeOfLastEventUseExisting>

<ns0:ProblemStateUseExisting>true</ns0:ProblemStateUseExisting>
                    <ns0:AlertHistory>
                        <ns0:AlertHistoryInsert>
                            <ns0:AlertId><xsl:value-of
select="a:ExternalEventID"/></ns0:AlertId>

<ns0:OwnerNameUseExisting>true</ns0:OwnerNameUseExisting>
                            <ns0:CommentUseExisting>false</ns0:CommentUseExisting>
                            <ns0:Comment>
                                Event has been updated in Oracle Enterprise
Manager
                                Occurred Date: <xsl:choose><xsl:when
test="normalize-space(a:SystemAttributes/a:OccurredDate) != ''"><xsl:value-of
select="a:SystemAttributes/a:OccurredDate"/></xsl:when><xsl:otherwise>N/A</xsl:oth
erwise></xsl:choose>
                                Reported Date: <xsl:choose><xsl:when
test="normalize-space(a:SystemAttributes/a:ReportedDate) != ''"><xsl:value-of
select="a:SystemAttributes/a:ReportedDate"/></xsl:when><xsl:otherwise>N/A</xsl:oth
erwise></xsl:choose>
                                Message: <xsl:value-of
select="a:SystemAttributes/Message"/>
                                Changed 'Severity' to '<xsl:value-of
select="$MOMSeverity"/>'
                            </ns0:Comment>
                            <xsl:choose>
                                <xsl:when test="$MOMSeverity = 'Success'">

<ns0:ResolutionStateUseExisting>false</ns0:ResolutionStateUseExisting>
                                    <ns0:ResolutionState>255</ns0:ResolutionState>
                                </xsl:when>
                                <xsl:otherwise>

<ns0:ResolutionStateUseExisting>true</ns0:ResolutionStateUseExisting>
                                    <ns0:ResolutionState>0</ns0:ResolutionState>
                                </xsl:otherwise>
                            </xsl:choose>

<ns0:CustomField1UseExisting>true</ns0:CustomField1UseExisting>

<ns0:CustomField2UseExisting>true</ns0:CustomField2UseExisting>

<ns0:CustomField3UseExisting>true</ns0:CustomField3UseExisting>

<ns0:CustomField4UseExisting>true</ns0:CustomField4UseExisting>
```

```
<ns0:CustomField5UseExisting>true</ns0:CustomField5UseExisting>

<ns0:LastModifiedByUseExisting>true</ns0:LastModifiedByUseExisting>

<ns0:TicketIDUseExisting>true</ns0:TicketIDUseExisting>
                        </ns0:AlertHistoryInsert>
                    </ns0:AlertHistory>
                </ns0:AlertUpdate>
            </ns0:updatedAlerts>
        </ns0:UpdateAlerts>
    </xsl:template>
</xsl:transform>
```

Example C–10 updateEvent_response.xsl

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:transform version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns:a="http://www.microsoft.com/EnterpriseManagement/Mom/Connector/V2"
               xmlns:ns0="http://xmlns.oracle.com/sysman/connector">
 <xsl:template match="a:UpdateAlertsResponse">
  <ns0:EMEventResponse>
   <xsl:choose>
    <xsl:when test="a:UpdateAlertsResult/a:UpdatedAlerts/a:guid">
     <ns0:SuccessFlag>true</ns0:SuccessFlag>
     <ns0:ExternalEventId>
      <xsl:value-of select="a:UpdateAlertsResult/a:UpdatedAlerts/a:guid"/>
     </ns0:ExternalEventId>
    </xsl:when>
    <xsl:otherwise>
     <ns0:SuccessFlag>false</ns0:SuccessFlag>
     <ns0:ErrorMessage>
      <xsl:value-of
select="a:UpdateAlertsResult/a:FailedUpdatedAlerts/a:FailedAlertUpdate/a:Error"/>
     </ns0:ErrorMessage>
    </xsl:otherwise>
   </xsl:choose>
  </ns0:EMEventResponse>
 </xsl:template>
</xsl:transform>
```

Example C–11 connectorDeploy.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns="http://xmlns.oracle.com/sysman/connector"
            targetNamespace="http://xmlns.oracle.com/sysman/connector"
            elementFormDefault="qualified">
  <xsd:include schemaLocation="connectorCommon.xsd"/>
  <xsd:element name="ManagementConnector">
    <xsd:annotation>
      <xsd:documentation>Deployment Descriptor for Management
Connectors</xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="Name" type="StringT64">
          <xsd:annotation>
              <xsd:documentation>
                The name of the connector type.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="Version" type="VersionT">
          <xsd:annotation>
              <xsd:documentation>
                Version of the connector type.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="EMCompatibleVersion" type="VersionT">
          <xsd:annotation>
              <xsd:documentation>
                The EM compabilitity version of the connector type.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="Description" type="StringT256">
          <xsd:annotation>
              <xsd:documentation>
                The description of the connector type.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="Category">
          <xsd:annotation>
              <xsd:documentation>
                The category of the connector type. It must be one of the three
                values listed next.
              </xsd:documentation>
          </xsd:annotation>
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:enumeration value="EventConnector"/>
              <xsd:enumeration value="TicketingConnector"/>
              <xsd:enumeration value="ChangeManagementConnector"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <!-- NewTargetType is for EventConnector only. -->
        <xsd:element name="NewTargetType" minOccurs="0">
          <xsd:annotation>
              <xsd:documentation>
                New target type definition for event connectors. This target type
                will be registered with Enterprise Manager and target instances
can
                be created subsequently, including a default target. These targets
                are used to accommodate external events.
              </xsd:documentation>
          </xsd:annotation>
          <xsd:complexType>
            <xsd:sequence>
           <xsd:element name="TargetTypeName" type="StringStrictT64">
                <xsd:annotation>
                    <xsd:documentation>
                      The name of the target type.
                    </xsd:documentation>
                </xsd:annotation>
              </xsd:element>
          <xsd:element name="TargetTypeDisplayName" type="StringT128">
                <xsd:annotation>
```

```
          <xsd:documentation>
           The name of the target type, as shown on UI.
          </xsd:documentation>
        </xsd:annotation>
       </xsd:element>
    <xsd:element name="DefaultTargetName" type="StringStrictT256">
        <xsd:annotation>
          <xsd:documentation>
           The name of the default target of the target type. The
default
           target will be used as a generic bucket to hold external
events.
          </xsd:documentation>
        </xsd:annotation>
       </xsd:element>
    <xsd:element name="DefaultTargetDisplayName" type="StringT256">
        <xsd:annotation>
          <xsd:documentation>
           The name of the default target of the target type, to be
displayed
            on UI.
          </xsd:documentation>
        </xsd:annotation>
       </xsd:element>
      </xsd:sequence>
     </xsd:complexType>
    </xsd:element>
    <xsd:element name="SOAPHeaderAuthentication"
         type="SOAPHeaderAuthenticationType" minOccurs="0">
     <xsd:annotation>
       <xsd:documentation>
        Specification for SOAP Header authentication.
       </xsd:documentation>
     </xsd:annotation>
    </xsd:element>
    <xsd:element name="HTTPBasicAuthentication"
       type="UsernamePasswordAuthenticationType" minOccurs="0">
     <xsd:annotation>
       <xsd:documentation>
        Specification for HTTP basic authentication.
       </xsd:documentation>
     </xsd:annotation>
    </xsd:element>
    <xsd:element name="UserNameTokenAuthentication"
       type="UsernamePasswordAuthenticationType" minOccurs="0">
     <xsd:annotation>
       <xsd:documentation>
        Specification for Username Token authentication.
       </xsd:documentation>
     </xsd:annotation>
    </xsd:element>
    <xsd:element name="ConfigVariable" type="ConfigVariableType"
          minOccurs="0" maxOccurs="20">
     <xsd:annotation>
       <xsd:documentation>
        The vaiables used during connector configuration. These variables
        are required by external system to complete connector
configuration,
        which includes regitering with the external system. For instance,
        one configuration variable can be the resolution state required by
```

```
                Microsoft Operation Manager.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="ConnectivityTestVariable" type="ConfigVariableType"
                     minOccurs="0">
          <xsd:annotation>
              <xsd:documentation>
                An optional variable used to test connection to an external
server.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      <xsd:element name="Service" type="ServiceType" maxOccurs="20">
          <xsd:annotation>
              <xsd:documentation>
                Specification for web services, which define how connector
framework
                can communicate with external system.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
    <xsd:element name="ExternalURL" type="ExternalURLType" minOccurs="0">
          <xsd:annotation>
              <xsd:documentation>
                Secification for the URL link to the external server, including
                the URL pattern and server specific variables. It is used to
provide links
                to external server for viewing ticket details.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="TemplateRegistration" type="TemplateRegistrationType"
                     minOccurs="0" maxOccurs="50">
          <xsd:annotation>
              <xsd:documentation>
                Specification for template registration. A template is registered
                based on the information provided in the element. A connector
deployment
                descriptor can have an optional list of upto 50 template
registratin
                elements.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="ServiceType">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a complext type for a web service.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="Method">
          <xsd:annotation>
              <xsd:documentation>
                The name of the web service method. Each connector category has a
                predefined set of methods as defined next.
              </xsd:documentation>
          </xsd:annotation>
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <!-- event connector: -->
            <xsd:enumeration value="setup"/>
            <xsd:enumeration value="initialize"/>
            <xsd:enumeration value="getNewAlerts"/>
            <xsd:enumeration value="getUpdatedAlerts"/>
            <xsd:enumeration value="acknowledgeAlerts"/>
            <xsd:enumeration value="updateAlerts"/>
            <xsd:enumeration value="createEvent"/>
            <xsd:enumeration value="updateEvent"/>
            <xsd:enumeration value="uninitialize"/>
            <xsd:enumeration value="cleanup"/>
            <!-- ticketing connector: -->
            <xsd:enumeration value="createTicket"/>
            <xsd:enumeration value="updateTicket"/>
            <xsd:enumeration value="getTicket"/>
            <!-- change management connector: -->
            <xsd:enumeration value="publishCS"/>
            <xsd:enumeration value="updateChangeRequest"/>
            <xsd:enumeration value="getChangeRequest"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="WebServiceEndpoint" type="StringT256">
          <xsd:annotation>
              <xsd:documentation>
                The web service end point indicating a specific location for
accessing
                a service.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
     <xsd:element name="SOAPAction" type="StringT64" minOccurs="0">
          <xsd:annotation>
              <xsd:documentation>
                The SOAP action which carries out the web service call for the
method.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      <xsd:element name="SOAPBindingType" minOccurs="0">
          <xsd:annotation>
              <xsd:documentation>
                The type of SOAP over HTTP binding. Choose from one of the four
                options defined next.
              </xsd:documentation>
          </xsd:annotation>
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="SOAP11HTTP_BINDING"/>
            <xsd:enumeration value="SOAP12HTTP_BINDING"/>
            <xsd:enumeration value="SOAP11HTTP_MTOM_BINDING"/>
            <xsd:enumeration value="SOAP12HTTP_MTOM_BINDING"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
    </xsd:sequence>
```

```
  </xsd:complexType>
  <xsd:complexType name="SOAPHeaderAuthenticationType">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a complext type for SOAP Header Authentication.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="Username" type="ConfigVariableType">
        <xsd:annotation>
            <xsd:documentation>
              The username of the authentication.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="Password" type="ConfigVariableType">
        <xsd:annotation>
            <xsd:documentation>
              The password of the authentication.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="AuthVariable" type="ConfigVariableType" minOccurs="0"
                   maxOccurs="20">
        <xsd:annotation>
            <xsd:documentation>
              An optional list of extra authentication variables besides username
              and password.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="SOAPHeader" type="StringT256">
        <xsd:annotation>
            <xsd:documentation>
              A SOAP header string serving as template for the SOAP header. It is
              to be updated with user inputs for variables defined above and
              bound with a HTTP request.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="UsernamePasswordAuthenticationType">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a complext type for Username Password
authentication.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="Username" type="ConfigVariableType">
        <xsd:annotation>
            <xsd:documentation>
              The username of the authentication.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="Password" type="ConfigVariableType">
        <xsd:annotation>
            <xsd:documentation>
```

```
              The password of the authentication.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="ConfigVariableType">
    <xsd:annotation>
        <xsd:documentation>
          This seciton defines a complext type for configuration variables.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="VariableName" type="StringStrictT32">
        <xsd:annotation>
            <xsd:documentation>
              Name of the variable.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="DisplayName" type="StringT64">
        <xsd:annotation>
            <xsd:documentation>
              Name of the variable used for display on UI.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
   <xsd:attribute name="required" type="xsd:boolean" default="false">
      <xsd:annotation>
          <xsd:documentation>
            A Flag indicating whether or not the variable is mandatory.
          </xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
  <xsd:complexType name="ExternalURLType">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a complex type for external URL.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="Pattern" type="StringT256">
          <xsd:annotation>
              <xsd:documentation>
                The URL pattern used to format links to the external server.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      <xsd:element name="ConfigVariable" type="ConfigVariableType" minOccurs="0"
                   maxOccurs="50">
          <xsd:annotation>
              <xsd:documentation>
                An optional list of configuration variablees representing the
details
                of the external server. They are used for constructing links to
                the server based on the URL pattern.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="TemplateRegistrationType">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a complex type for template registration metadata
          which is used to register templates during connector deployment.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="FileName" type="StringT256">
          <xsd:annotation>
              <xsd:documentation>
                The template file name.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      <xsd:element name="InternalName" type="StringStrictT128">
          <xsd:annotation>
              <xsd:documentation>
                A name representing the template in the connector framework.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      <xsd:element name="TemplateName" type="StringStrictT128">
          <xsd:annotation>
              <xsd:documentation>
                The template display name to be used on UI.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      <xsd:element name="TemplateType">
          <xsd:annotation>
              <xsd:documentation>
                The template type as one of the three options defined next.
              </xsd:documentation>
          </xsd:annotation>
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="InboundXSL"/>
            <xsd:enumeration value="OutboundXSL"/>
            <xsd:enumeration value="OutboundXML"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="Description" type="StringT512">
          <xsd:annotation>
              <xsd:documentation>
                A description of the template.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```

Example C–12 EMEvent.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns="http://xmlns.oracle.com/sysman/connector"
            targetNamespace="http://xmlns.oracle.com/sysman/connector"
            elementFormDefault="qualified"
xmlns:jaxb="http://java.sun.com/xml/ns/jaxb"
  jaxb:version="2.0"
  xmlns:xjc="http://java.sun.com/xml/ns/jaxb/xjc"
  jaxb:extensionBindingPrefixes="xjc">

  <xsd:annotation>
    <xsd:appinfo>
      <jaxb:globalBindings>
        <xjc:simple />
      </jaxb:globalBindings>
    </xsd:appinfo>
  </xsd:annotation>
   <xsd:include schemaLocation="connectorCommon.xsd"/>
   <xsd:element name="EMEvent" type="EMEventType" />
   <xsd:complexType name="EMEventType">
         <xsd:sequence>
            <xsd:element name="ConnectorGUID" type="xsd:string"/>
            <xsd:element name="ExternalEventID" type="xsd:string"
                         minOccurs="0"/>
            <xsd:element name="NotificationRuleOwner" type="xsd:string"/>
            <xsd:element name="NotificationRuleName" type="xsd:string"/>
            <xsd:element name="ConnectorVariable" type="VariableType"
                         minOccurs="0" maxOccurs="50"/>
            <xsd:element name="Property" type="PropertyType" minOccurs="0"
                         maxOccurs="50"/>
            <xsd:element name="SystemAttributes"
                         type="EventSystemAttributesType"/>
            <xsd:element name="EventClassSpecificAttributes">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:choice minOccurs="0" maxOccurs="200">
                     <xsd:element name="StringAttribute" type="StringValueType"/>
                     <xsd:element name="NumberAttribute" type="StringValueType"/>
                     <xsd:element name="RawAttribute" type="StringValueType"/>
                     <xsd:element name="DateAttribute" type="DateValueType"/>
                     </xsd:choice>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
            <xsd:element name="EventContextAttributes">
               <xsd:complexType>
                  <xsd:sequence>
                     <xsd:choice minOccurs="0" maxOccurs="200">
                     <xsd:element name="StringAttribute" type="StringValueType"/>
                     <xsd:element name="NumberAttribute" type="StringValueType"/>
                     </xsd:choice>
                  </xsd:sequence>
               </xsd:complexType>
            </xsd:element>
         </xsd:sequence>
      </xsd:complexType>
   <xsd:complexType name="EventSystemAttributesType">
      <xsd:sequence>
         <xsd:element name="EventClass" type="xsd:string"/>
         <xsd:element name="EventID" type="xsd:string"/>
         <xsd:element name="SequenceID" type="xsd:string"/>
         <xsd:element name="OccurredDate" type="xsd:dateTime" minOccurs="0"/>
         <xsd:element name="ReportedDate" type="xsd:dateTime"/>
         <xsd:element name="DisplayTZ" type="xsd:string"/>
         <xsd:element name="EventName" type="xsd:string"/>
         <xsd:element name="Severity" type="xsd:string"/>
         <xsd:element name="SeverityCode" type="xsd:string"/>
         <xsd:element name="SourceInfo" type="SourceInfoType"/>
         <xsd:element name="Message" type="xsd:string" minOccurs="0"/>
         <xsd:element name="ActionMessage" type="xsd:string" minOccurs="0"/>
         <xsd:element name="EventURL" type="xsd:string"/>
         <xsd:element name="AutoClose" type="xsd:boolean"/>
         <xsd:element name="EventCategory" type="xsd:string" minOccurs="0"
                      maxOccurs="50"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="StringValueType">
      <xsd:sequence>
         <xsd:element name="Name" type="xsd:string"/>
         <xsd:element name="Value" type="xsd:string"/>
      </xsd:sequence>
   </xsd:complexType>
   <xsd:complexType name="DateValueType">
      <xsd:sequence>
         <xsd:element name="Name" type="xsd:string"/>
         <xsd:element name="Value" type="xsd:dateTime"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:schema>
```

Example C–13 connectorCommon.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns="http://xmlns.oracle.com/sysman/connector"
            targetNamespace="http://xmlns.oracle.com/sysman/connector"
            elementFormDefault="qualified">
  <xsd:include schemaLocation="externalEvent.xsd"/>
  <xsd:complexType name="SourceInfoType">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a complext type for Source Information.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
   <xsd:element name="SourceObjInfo" type="SourceObjInfoType" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>
              This element defines the data structure for the source object, the
EM
              subsystem or component, that raises an EM event or an incident.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    <xsd:element name="TargetInfo" type="TargetInfoType" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>
              The element defines the data structure for an EM target as related
              to the connector framework.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="SourceObjInfoType">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a complex type for Source Object Information.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="ObjID" type="xsd:string">
        <xsd:annotation>
            <xsd:documentation>
              The unique ID to identify the source object.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="ObjName" type="xsd:string">
        <xsd:annotation>
            <xsd:documentation>
              The name of the source object.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
     <xsd:element name="ObjOwner" type="xsd:string" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>
              The owner of the source object.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="SourceObjType" type="xsd:string">
        <xsd:annotation>
            <xsd:documentation>
              The type of the source object.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
   <xsd:element name="SourceObjSubType" type="xsd:string" minOccurs="0">
        <xsd:annotation>
            <xsd:documentation>
              The subtype of the source object.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="TargetInfoType">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a complex type for target information.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="TargetGUID" type="xsd:string">
        <xsd:annotation>
            <xsd:documentation>
              A unique GUID for the target.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="TargetName" type="xsd:string">
        <xsd:annotation>
            <xsd:documentation>
              Name of the target.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="TargetType" type="xsd:string">
        <xsd:annotation>
            <xsd:documentation>
              Type of the target.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="TargetTypeLabel" type="xsd:string">
        <xsd:annotation>
            <xsd:documentation>
              The display label of the target type.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="TargetURL" type="xsd:string">
        <xsd:annotation>
            <xsd:documentation>
              The URL of the target.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="TargetProperty" type="PropertyType" minOccurs="0"
                   maxOccurs="50">
        <xsd:annotation>
            <xsd:documentation>
              An optional list of properties for the target.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="PropertyType">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a complex type for a property attribute.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="Name" type="xsd:string">
          <xsd:annotation>
              <xsd:documentation>
                A string name defining a property attribute.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
     <xsd:element name="Value" type="xsd:string" nillable="true">
          <xsd:annotation>
              <xsd:documentation>
                A non-null string value.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="VariableType">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a complex type for a general variable.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="VariableName" type="StringStrictT32">
          <xsd:annotation>
              <xsd:documentation>
                Name of the variable. It has to be a string containing 1 or upto
                32 upper case or lower case letters or numbers.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      <xsd:element name="VariableValue" type="StringT2048">
          <xsd:annotation>
              <xsd:documentation>
                Value of the variable. It has to be a string containing 1 or upto
                2048 characters.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="GetAlertsResponse">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a complex type for responses to a getAlerts
request.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="Alert" minOccurs="0" maxOccurs="200">
          <xsd:annotation>
              <xsd:documentation>
                The individual alerts contained in the response. A response may
have
                upto 200 alerts.
              </xsd:documentation>
          </xsd:annotation>
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element ref="ExternalEvent">
              <xsd:annotation>
                  <xsd:documentation>
                    Details of the external event in the alert, as defined in
                    ExternalEvent.xsd.
                  </xsd:documentation>
              </xsd:annotation>
            </xsd:element>
            <xsd:element name="InstanceVariable" type="VariableType"
                         minOccurs="0" maxOccurs="50">
              <xsd:annotation>
                  <xsd:documentation>
                    A list of instance variables for the alert.
                  </xsd:documentation>
              </xsd:annotation>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="ConnectorVariablesType">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a complex type for connector variables. An element
          of type ConnectorVariablesType may have up to 50 connector variables, as
          defined next.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="ConnectorVariable" type="VariableType" minOccurs="0"
                   maxOccurs="50">
        <xsd:annotation>
            <xsd:documentation>
              A connector varialbe as a name/value pair.
            </xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:simpleType name="StringT64">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a simple type for a String with maximum length of
          64 bytes.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="64"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="StringT128">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a simple type for a String with maximum length of
          128 bytes.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="128"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="StringT256">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a simple type for a String with maximum length of
          256 bytes.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
```

```
      <xsd:maxLength value="256"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="StringT512">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a simple type for a String with maximum length of
          512 bytes.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="512"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="StringT2048">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a simple type for a String with maximum length of
          2048 bytes.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="2048"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="StringStrictT16">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a simple type for a String with maximum length of
          16 bytes. The String can only contain lower or upper case letters,
numbers,
          and the underscore characters.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="16"/>
      <xsd:pattern value="([a-zA-Z0-9_])*"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="StringStrictT32">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a simple type for a String with maximum length of
          32 bytes. The String can only contain lower or upper case letters,
numbers,
          and the underscore characters.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="32"/>
      <xsd:pattern value="([a-zA-Z0-9_])*"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="StringStrictT64">
    <xsd:annotation>
        <xsd:documentation>
```

```
          This section defines a simple type for a String with maximum length of
          64 bytes. The String can only contain lower or upper case letters,
numbers,
          and the underscore characters.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="64"/>
      <xsd:pattern value="([a-zA-Z0-9_])*"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="StringStrictT128">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a simple type for a String with maximum length of
          128 bytes. The String can only contain lower or upper case letters,
numbers,
          and the underscore characters.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="128"/>
      <xsd:pattern value="([a-zA-Z0-9_])*"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="StringStrictT256">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a simple type for a String with maximum length of
          256 bytes. The String can only contain lower or upper case letters,
numbers,
          and the underscore characters.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="256"/>
      <xsd:pattern value="([a-zA-Z0-9_])*"/>
    </xsd:restriction>
  </xsd:simpleType>
  <xsd:simpleType name="VersionT">
    <xsd:annotation>
        <xsd:documentation>
          This section defines a simple type for a String with maximum length of
          20 bytes. The String can only contain numbers and the period characters.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="xsd:string">
      <xsd:minLength value="1"/>
      <xsd:maxLength value="20"/>
      <xsd:pattern value="([0-9.])*"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:schema>
```

Example C–14 EMEventResponse.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns="http://xmlns.oracle.com/sysman/connector"
            targetNamespace="http://xmlns.oracle.com/sysman/connector"
            elementFormDefault="qualified">
  <xsd:include schemaLocation="../common/connectorCommon.xsd"/>
  <xsd:element name="EMEventResponse">
    <xsd:annotation>
        <xsd:documentation>
          The response from external server for an EM event it has received.
        </xsd:documentation>
    </xsd:annotation>
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="SuccessFlag" type="xsd:boolean">
          <xsd:annotation>
              <xsd:documentation>
                The flag to indicate whether or not the event has been
successfully
                 inserted or updated at the external system.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        <xsd:choice>
          <xsd:element name="ExternalEventId" type="StringT128">
            <xsd:annotation>
                <xsd:documentation>
                  The ID to identify the event created in the external event
system.
                  It is returned by the external system when the event is
successfully
                  inserted or updated.
                </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="ErrorMessage" type="StringT2048">
          <xsd:annotation>
              <xsd:documentation>
                The error message returned by the external system when the event
                fails to be inserted or updated.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
        </xsd:choice>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Example C–15 setupResponse.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns="http://xmlns.oracle.com/sysman/connector"
            targetNamespace="http://xmlns.oracle.com/sysman/connector"
            elementFormDefault="qualified">

  <xsd:include schemaLocation="../common/connectorCommon.xsd"/>
 <xsd:element name="InitializeResponse" type="ConnectorVariablesType">
```

```
          <xsd:annotation>
              <xsd:documentation>
                The response for an initialize reqeust. It contains a list of
                connector variables, which are name/value pairs.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>

</xsd:schema>
```

Example C–16 initializeResponse.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns="http://xmlns.oracle.com/sysman/connector"
            targetNamespace="http://xmlns.oracle.com/sysman/connector"
            elementFormDefault="qualified">

  <xsd:include schemaLocation="../common/connectorCommon.xsd"/>
 <xsd:element name="InitializeResponse" type="ConnectorVariablesType">
          <xsd:annotation>
              <xsd:documentation>
                The response for an initialize reqeust. It contains a list of
                connector variables, which are name/value pairs.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>

</xsd:schema>
```

Example C–17 uninitialize_response.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
            xmlns="http://xmlns.oracle.com/sysman/connector"
            targetNamespace="http://xmlns.oracle.com/sysman/connector"
            elementFormDefault="qualified">

  <xsd:include schemaLocation="../common/connectorCommon.xsd"/>
 <xsd:element name="UninitializeResponse" type="ConnectorVariablesType">
          <xsd:annotation>
              <xsd:documentation>
                The response for an uninitialize reqeust. It contains a list of
                connector variables, which are name/value pairs.
              </xsd:documentation>
          </xsd:annotation>
        </xsd:element>

</xsd:schema>
```

Glossary

This glossary defines the different event types

Event Name	Type	Description
Name	StringT64	The name of the connector type
Version	VersionT	The version of the connector type
EMCompatibleVersion	VersionT	The EM compatibility version of the connector type
Description	StringT256	The description of the connector type
Category	-	The category of the connector type. It must be one of the three values listed next.
NewTargetType	-	New target type definition for event connectors. This target type will be registered with Enterprise Manager and target instances can be created subsequently, including a default target. These targets are used to accommodate external events.
TargetTypeName	StringStrictT64	The name of the target type
TargetTypeDisplayName	StringT128	The name of the target type, as shown on UI
DefaultTargetName	StringStrictT256	The name of the default target of the target type. The default target will be used as a generic bucket to hold external events.
DefaultTargetDisplayName	StringT256	The name of the default target of the target type, to be displayed on UI.
SOAPHeaderAuthentication	SOAPHeaderAuthenticationType	Specification for SOAP Header authentication.
HTTPBasicAuthentication	UsernamePasswordAuthenticationType	Specification for HTTP basic authentication.
UserNameTokenAuthentication	UsernamePasswordAuthenticationType	Specification for Username Token authentication.
ConfigVariable	ConfigVariableType	The variables used during connector configuration. These variables are required by external system to complete connector configuration, which includes registering with the external system. For instance, one configuration variable can be the resolution state required by Microsoft Operation Manager.

Event Name	Type	Description
ConnectivityTestVariable	ConfigVariableType	An optional variable used to test connection to an external server.
Service	ServiceType	Specification for web services, which define how connector framework can communicate with external system.
ExternalURL	ExternalURLType	Specification for the URL link to the external server, including the URL pattern and server specific variables. It is used to provide links to external server for viewing ticket details.
TemplateRegistration	TemplateRegistrationType	Specification for template registration. A template is registered based on the information provided in the element. A connector deployment descriptor can have an optional list of up to 50 template registration elements.
Method	-	The name of the web service method. Each connector category has a predefined set of methods as defined next.
WebServiceEndpoint	StringT256	The web service end point indicating a specific location for accessing a service.
SOAPAction	StringT64	The SOAP action which carries out the web service call for the method
SOAPBindingType	-	The type of SOAP over HTTP binding. Choose from one of the four options defined next
Username	ConfigVariableType	The username of the authentication.
Password	ConfigVariableType	The password of the authentication
AuthVariable	ConfigVariableType	An optional list of extra authentication variables besides username and password
SOAPHeader	StringT256	A SOAP header string serving as template for the SOAP header. It is to be updated with user inputs for variables defined above and bound with a HTTP request.
VariableName	StringStrictT32	Name of the variable
DisplayName	StringT64	Name of the variable used for display on UI.
Pattern	StringT256	The URL pattern used to format links to the external server.
FileName	StringT256	The template file name
InternalName	StringStrictT128	A name representing the template in the connector framework
TemplateName	StringStrictT128	The template display name to be used on UI.
TemplateType	-	The template type as one of the three options defined next.
Description	StringT512	A description of the template

Index

A

Add Target Page, data exchange connectors, 3-7
architecture of data exchange connectors, 3-1
auto ticketing, help desk connectors, 1-1

C

Connector Framework
  event connectors, 2-1

D

data exchange
  OBAM artifacts for inbound session, 3-39
data exchange connectors, 3-38
  Add Target Page, 3-7
  architecture, 3-1
  checking logs, 3-44
  creating a data exchange hub, 3-6
  data exchange hub, 3-3
  data flow tips, 3-43
  denormalized message format, 3-4
  EM-BAM data objects, 3-34
  EM-BAM EMS definitions, 3-35
  importing OBAM artifacts, 3-34
  inbound alert schema, 3-33
  inbound indicators schema, 3-31
  inbound JMS topics, 3-31
  inbound message schemas, 3-31
  JNDI details, 3-43
  message example defaults, 3-32
  normalized message format, 3-4
  notification method, 3-43
  notification rules, 3-43
  qualified XML message sample, 3-31
  Select Business Events/Indicators Page, 3-28
  Session Setup Page, 3-7, 3-28
  setting up, 3-5
  setting up data flow, 3-34
  unqualified XML message sample, 3-31
  updating JNDI, 3-38
denormalized message format, outbound message schema, 3-22

E

EM-BAM
  data objects, 3-34, 3-38
  EMS definitions, 3-35
  enterprise link plans, 3-38
EM-BAM data objects, 3-38
EM-BAM enterprise link plans, 3-38
event connectors
  metadata files required, 2-1

H

help desk connectors
  auto ticketing, 1-1
  manual ticketing, 1-1
  metadata files, 1-1

I

importing OBAM artifacts, 3-34
inbound alert schema, 3-33
inbound indicators schema, 3-31

J

JNDI details, data exchange connectors, 3-43

M

manual ticketing, help desk connectors, 1-1
message example defaults, 3-32
metadata files
  help desk connectors, 1-1
  required for event connectors, 2-1

N

normalized alert message, outbound message schema, 3-17
normalized message format, outbound message schema, 3-13
normalized metric data message, outbound message schema, 3-16
normalized metric message, outbound message schema, 3-14
normalized security filter message, outbound message schema, 3-15
normalized target message, outbound message schema, 3-13
notification method, data exchange connectors, 3-43
notification rules
  data exchange connectors, 3-43

O

OBAM
  artifacts for inbound session, 3-39
  EM-BAM data objects, 3-34, 3-38
  EM-BAM EMS definitions, 3-35
  EM-BAM enterprise links plans, 3-38
  importing artifacts, 3-34
  setting up data flow, 3-34
  updating JNDI, 3-38
outbound message schema
  denormalized message format, 3-22
  normalized alert message, 3-17
  normalized message format, 3-13
  normalized metric data message, 3-16
  normalized metric message, 3-14
  normalized security filter message, 3-15
  normalized target message, 3-13

Q

qualified XML message sample, 3-31

S

Select Business Events/Indicators Page, data exchange connectors, 3-28
Session Setup Page, data exchange connector, 3-7, 3-28

U

unqualified XML message sample, 3-31
updating JNDI, 3-38