Oracle® Fusion Middleware

Healthcare Integration User's Guide for Oracle SOA Suite

12*c* (12.1.3)

E39963-06

January 2016

Documentation for developers that describes how to create and configure Oracle SOA Suite for healthcare integration applications that map and transmit information between disparate healthcare systems. This document also describes how to monitor and manage deployed Oracle Healthcare applications.

ORACLE®

Oracle Fusion Middleware Healthcare Integration User's Guide for Oracle SOA Suite, 12c (12.1.3)

E39963-06

Contents

6 Working with Mapsets

7 Working with Internal Delivery Channels

8 Working with Dashboards

9 Working with Reports

13   Configuring System Settings

14   Provisioning Users

15   Enabling Web-Service-Based Message Exchange in Oracle Healthcare

16   Oracle Healthcare Command-Line Tools

F   Batching HL7 Messages

G   Configuration for Functional Acknowledgment 999

H   TA1/999 Generation on Error for HIPAA Documents

I   Implementing SNIP Validation in HIPAA

J   Improving Endpoint Scalability by Using NIO

Preface

This guide describes how to use Oracle SOA Suite for healthcare integration.

Audience

Oracle Fusion Middleware Healthcare Integration User's Guide for Oracle SOA Suite is intended for healthcare organizations that exchange healthcare data, and want to design, deploy, monitor, and manage healthcare integration applications.

Related Documents

For information about Oracle SOA Suite products, see the following:

- *Oracle Fusion Middleware User's Guide for Oracle B2B*

- *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*

- *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*

- *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite and Oracle Business Process Management Suite*

- *Oracle Fusion Middleware User's Guide for Technology Adapters*

- *Oracle Fusion Middleware Performance and Tuning Guide*

For information about the Java API documentation (Javadoc), see the following:

- *Oracle Fusion Middleware B2B Callout Java API Reference*

- *Oracle Fusion Middleware B2B Instance Message Java API Reference*

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.

Convention	Meaning
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

What's New in This Guide for Release 12.1.3

This guide has been updated in several ways. The following table lists the sections that have been added or changed.

For a list of known issues (release notes), see the "Known Issues for Oracle SOA Products and Oracle AIA Foundation Pack" at `http://www.oracle.com/technetwork/middleware/docs/soa-aiafp-knownissuesindex-364630.html`.

Sections	Changes Made	May 2014
Chapter 4 Working with Endpoints		
Enabling SSL/TLS Support for MLLP Endpoints	Section added that describes how to enable SSL/TLS support for secure messaging.	X
Handling Actionable Errors for an MLLP Endpoint	Section added that describes SOA Suite for healthcare integration support for receiving and delivering HL7 v2.x Messages over the MLLP 1.0 protocol and handling issues such as message parse failure, message validation failure, or rejection.	X
Overriding Document Parameters at the Endpoint Level	Section describes how to override document parameters at individual endpoint level.	X
Healthcare and Oracle Managed File Transfer Integration	Section describes how Oracle Healthcare recognizes Oracle Managed File Transfer (MFT) as a remote endpoint. In endpoint configurations, MFT is added as additional transport protocol. Oracle Healthcare uses an outbound endpoint to send files to MFT and an inbound endpoint to receive files.	X
Chapter 9 Working with Reports		
Viewing a Business Message Instance	Section describes new features that enable end-to-end monitoring.	X
End-to-End Monitoring of Runtime Data	Section describes Reports page end-to-end monitoring for fan-out use cases.	X
Chapter 15 Enabling Web-Service-Based Message Exchange in Oracle Healthcare		
Attaching security policies	Section describes how to attach security policies to an endpoint.	X

Sections	Changes Made	May 2014
Various New Appendices		
Interface Sequencing	This appendix discusses how the Oracle SOA Suite for healthcare integration achieves end-to-end First In First Out (FIFO) processing of messages by using Interface Sequencing.	X
TA1/999 Generation on Error for HIPAA Documents	This appendix discusses how to implement the Strategic National Implementation Process (SNIP) validation for HIPAA messages.	X
Implementing SNIP Validation in HIPAA	This appendix discusses how to implement the Strategic National Implementation Process (SNIP) validation for HIPAA messages.	X
Improving Endpoint Scalability by Using NIO	This appendix discusses how linear scalability of a large number of endpoints can be achieved by using the Java New I/O (NIO) API.	X
B2B and Healthcare Domain Topology Best Practices	This appendix describes best practices for deploying HL7 and X12 HIPPA interfaces in different domains.	X
Instance Tracking and Error Hospital Integration	This appendix describes instance tracking and error hospital tracking functionality.	X

1

Getting Started with Oracle SOA Suite for Healthcare Integration

This chapter provides an overview of the Oracle SOA Suite for healthcare integration, which provides a Web-based user interface for creating applications to transmit and transform data between various healthcare systems. The user interface also provides monitoring and management capabilities for the messages shared between those systems.

This chapter contains the following topics:

- Introduction to Oracle SOA Suite for Healthcare Integration

- Installing and Configuring Oracle SOA Suite for Healthcare Integration

- Logging in to Oracle SOA Suite for Healthcare Integration

- Using the Oracle SOA Suite for Healthcare Integration User Interface

- Accessibility Options

1.1 Introduction to Oracle SOA Suite for Healthcare Integration

Oracle SOA Suite for healthcare integration utilizes several features of Oracle SOA Suite to help you design, create, and manage applications that process healthcare data. Oracle SOA Suite for healthcare integration includes a Web-based user interface where you can create and configure healthcare integration applications, as well as monitor and manage the messages processed through those applications. You can also use the Oracle Document Editor to create and configure document definitions, which define message structures.

The Oracle SOA Suite for Healthcare integration user interface provides additional features, such as support for additional messaging protocols and the ability to create and manage endpoints, manage documents, create mapsets, and create Java callouts.

1.1.1 Oracle SOA Suite for Healthcare Integration Components

A healthcare integration application includes several components that define how messages are received, processed, and transmitted. The components that make up a healthcare integration application include the following. Document definitions and endpoints are required components for a healthcare integration application.

- **Document Definitions**: Use document definitions to define the structure of the messages that are used in a healthcare transaction. Predefined templates are provided for HL7 messages when you install the Oracle Document Editor. Document definitions are categorized by protocol, version, and type.

- **Endpoints**: An endpoint brings together all of the above components for one external system. The endpoint defines whether messages are being sent or received

by Oracle SOA Suite for healthcare integration, the transport protocol, acknowledgment handling, and whether messages are validated or translated. Endpoints are associated with document definitions and, optionally, internal delivery channels, callouts, mapsets, and SOA Suite composite applications.

- **Mapsets**: Use mapsets to define the transformation from one native format to another, bypassing the step of translating to XML and back. For example, a mapset can map fields from a HIPAA 4010 document directly to a HIPAA 5010 document. A mapset includes source and target document definitions, and a file that defines the mapping.

- **Callouts**: Use callouts to incorporate your own Java code into a healthcare integration application.

- **Internal Delivery Channels**: Use internal delivery channels when you want the healthcare integration application to receive messages from or send messages to a JMS topic or queue.

1.1.2 Management and Monitoring Tools

After you create and deploy a healthcare integration application, you can monitor and manage the messages processed through the application using the Oracle SOA Suite for healthcare integration Web-based user interface. The reporting feature allows you to view a list of messages processed and then to drill deeper into message content, protocol information, and so on. You can also resubmit messages or configure automatic retries, configure partner downtimes, and purge selected messages from the metadata repository.

The dashboard feature allows you to view a summary of messages sent for each running endpoint. You can also navigate deeper to view the volume of messages sent, received and errored for the specified time period. This information is further categorized by document type. You can create custom dashboards and reports based on a variety of criteria. Several default reports are already provided for you.

1.1.3 Oracle SOA Suite and Healthcare Integration

Oracle SOA Suite for healthcare integration provides SOA features and components that extend business processes to healthcare systems. When Oracle SOA Suite for healthcare integration is used in a SOA composite application, you can model an end-to-end business process integration.

Oracle SOA Suite provides a complete set of service infrastructure components for designing, deploying, and managing composite applications. The multiple technology components of a composite application share common capabilities, including a single deployment and management model and tooling, end-to-end security, and unified metadata management.

In a SOA implementation, Oracle SOA Suite for healthcare integration functions as a binding component, with network protocols and services that enable message sending and receiving. As a service (inbound), the SOA composite application receives messages from Oracle SOA Suite for healthcare integration. As a reference (outbound), the SOA composite application passes a message to Oracle SOA Suite for healthcare integration, which in turn sends the message to external systems.

For more information about Oracle SOA Suite, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

1.1.4 Oracle SOA Suite for Healthcare Integration Metadata

Oracle SOA Suite for healthcare integration instance data is stored and managed within the SOAINFRA schema of your database. Metadata for design-time and configuration is stored and managed through Metadata Services (MDS), available in Oracle Fusion Middleware. For more information about MDS, see *Oracle Fusion Middleware Administrator's Guide*

Due to the storage of healthcare integration data in the metadata repository, it is possible that the tablespace might become full. If this occurs, increase the size of the tablespace. Increasing the size of the redo log file also helps to improve performance when importing large configurations. A larger log file requires more space but reduces the necessity for applications to retry the operation.

1.1.5 Security

Security for Oracle SOA Suite for healthcare integration is handled through Oracle WebLogic Server. You can create user accounts in WebLogic Server for people who must view and modify components in Oracle SOA Suite for healthcare integration and then provision those accounts using the Oracle SOA Suite for healthcare integration console. In order to have edit privileges on the healthcare integration user interface, users must be granted either the Administrator or Monitor role in Oracle SOA Suite for healthcare integration.

1.2 Installing and Configuring Oracle SOA Suite for Healthcare Integration

Installation and configuration of Oracle SOA Suite for healthcare integration is described in detail in *Installing and Configuring B2B and Healthcare*.

1.2.1 Importing the HL7 DocType into the MDS

The following instructions describe how to import the HL7 DocType into the MDS.

> **Note:**
>
> Be sure to substitute paths in your installation environment for the *italic* portions of the paths shown in the following steps (for example, *ORACLE_HOME*).

1. Download the HL7 Healthcare libraries (`ofm_healthcare_lib_generic_12.1.3.0_disk1_1of1.zip`) from the Oracle Technology Network (OTN). Unzip the `ofm_healthcare_lib_generic_12.1.3.0_disk1_1of1.zip` file to extract the various versions of the HL7 2.x documents into the following directory:

    ORACLE_HOME`/soa/thirdparty/healthcare/hl7_doctypes`

2. In a command prompt window, set the `JAVA_HOME` and `ANT_HOME` environment variables as follows:

    `setenv ANT_HOME` *ORACLE_HOME*`/oracle_common/modules/ org.apache.ant_1.9.2`

    `setenv PATH=$ANT_HOME/bin:$PATH`

    `setenv JAVA_HOME` *ORACLE_HOME*/*jrockit_install_directory*

```
setenv PATH=$JAVA_HOME/bin:$PATH
```

3. Create the `jndi.properties` file.

   a. At a command prompt, make the following directory the current directory:

      ORACLE_HOME`/soa/bin`

   b. Run the following command:

      ```
      ant -f ant-hcfp-util.xml hcfpcreate-prop
      ```

   c. Edit the `jndi.properties` file to include the WebLogic password and make sure that the value of `portNumber` and other parameters are correct.

4. If you unzipped the healthcare libraries in the locations shown in step 1, you can import any HL7 v2.x DocType version by executing the `hcfpimport` Ant target to import the HL7 2.3.1 DocTypes as shown in the following example.

   ```
   ant -f ant-hcfp-util.xml hcfpimport -Dlocalfile=true -
   Dexportfile="ORACLE_HOME/soa/soa/thirdparty/healthcare/
   hl7_doctypes/HL7DocType-2.3.1.zip"
   ```

1.3 Logging in to Oracle SOA Suite for Healthcare Integration

These instructions assume that you have installed Oracle SOA Suite, which includes Oracle SOA Suite for Healthcare integration.

See *Oracle Fusion Middleware Installation Guide for Oracle SOA Suite and Oracle Business Process Management Suite* for more information.

Use a supported Web browser:

- Mozilla Firefox 2.0, 3.0 and 3.5 or later

- Apple Safari 3.2, 4.0, and 5.0

- Google Chrome 1.0 or later

- Microsoft Internet Explorer 8 and 9 (with Compatibility View turned off)

To log on to Oracle SOA Suite for healthcare integration:

1. Open a supported Web browser and go to:

   `http://host_name:port_number/healthcare`

   where:

   - `host_name` is the name of the host on which Oracle SOA Suite is installed. (In a cluster environment, the `host_name` can be the front end load balancer.)

   - `port_number` is the port number used by the Managed Server to listen for regular HTTP (non-SSL) or HTTPS (SSL) connections. (In a cluster environment, the `port_number` can be the router port.)

     See Finding Port Information, for more information.

   - `/healthcare` accesses the healthcare integration user interface.

2. On the log-in page, enter the following:

For This Field...	Do...
Username	Enter the default administrator user name.
Password	Use the administrator password from your Oracle Fusion Middleware installation.

3. Click **Login**.

1.3.1 Finding Port Information

You can find port number information in the following ways:

- From Oracle WebLogic Server Administration Console:

  1. Log on to the console.

  2. In the Domain Structure pane, shown in , expand **Environment** and click **Servers**.

Figure 1-1   Domain Structure Nodes in Oracle WebLogic Server Administration Console

3. Note the **Listen Port** column for your server.

- Or from *MW_HOME*/user_projects/domains/*your_domain_name*/config/config.xml

```
<server>
  <name>soa_server1</name>
  <ssl>
    <name>soa_server1</name>
    <listen-port>8002</listen-port>
  </ssl>
  <machine>LocalMachine</machine>
```

```
                    <listen-port>8001</listen-port>
                    <listen-address/>
                </server>
```

1.4 Using the Oracle SOA Suite for Healthcare Integration User Interface

Healthcare activities are grouped as follows:

- Designer

- Dashboards

- Reports

1.4.1 Designer

Use the **Configuration** subtab of the **Designer** tab, shown in Figure 1-2, to configure document protocols, endpoints, callouts, mapsets, and the Administration subtab to modify runtime and user interface settings and internal delivery channel settings, and to import, export, and purge metadata.

Figure 1-2   Designer Tab
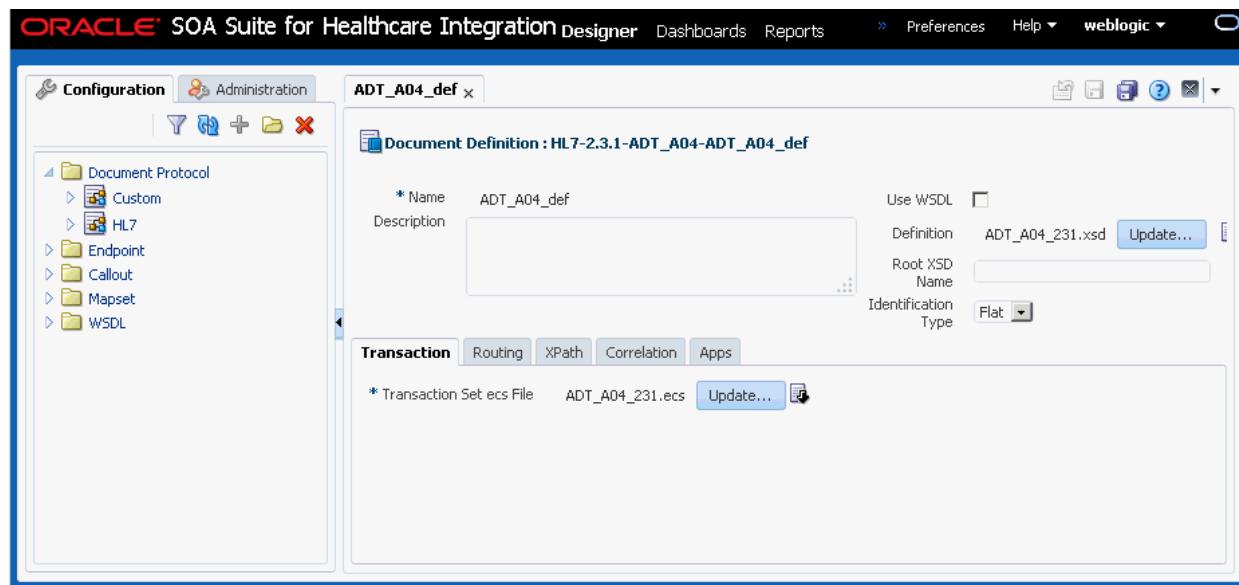
1.4.1.1 Configuring Documents

Using the **Document Protocol** folder in the **Configuration** tab, you can create, modify, or delete document protocols, document protocol version, document type, and document definition as shown in Figure 1-3. These document definitions can later be associated with endpoints.

Figure 1-3   The Document Protocol Folder
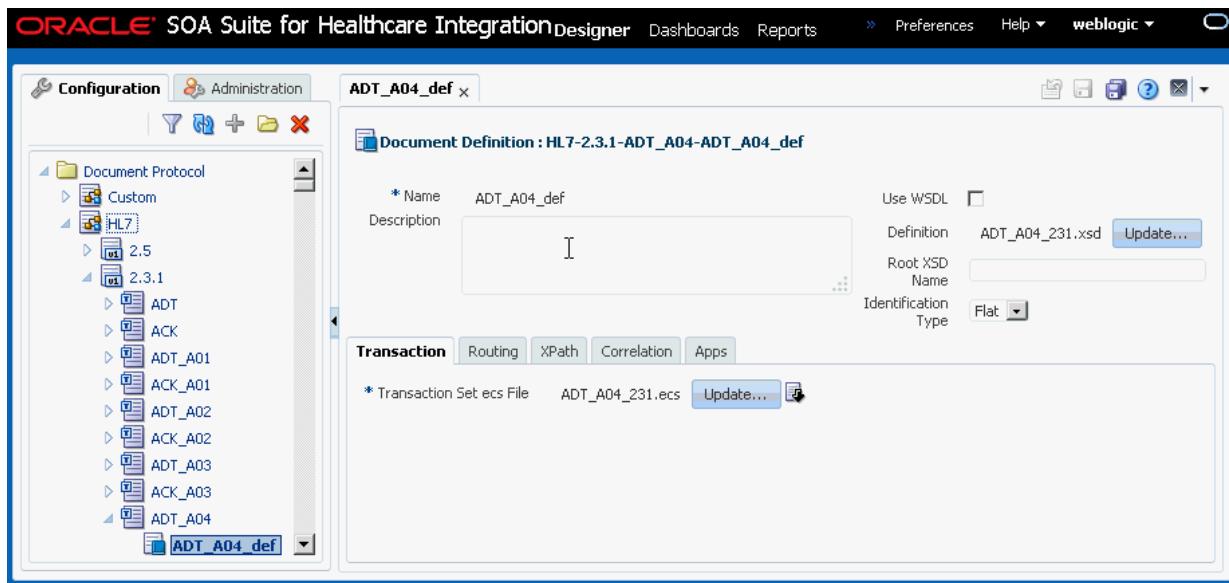
Note:

You can create documents with Oracle Document Editor, or you can use a pre-seeded export zip file that contains already created document definitions by using the Import/Export feature available in the Administration tab.

See Working with Document Types and Protocols for more information on documents and document protocols.

1.4.1.2 Configuring Endpoints

Endpoints are locations from where messages are sent or received. An endpoint can be a URL, folders, or path, among others. You can use the **Endpoints** folder in the **Configuration** tab to create, modify, or delete endpoints, as well as associate an endpoint with document definitions.

Figure 1-4 displays a sample endpoint.

Figure 1-4   Sample Endpoint
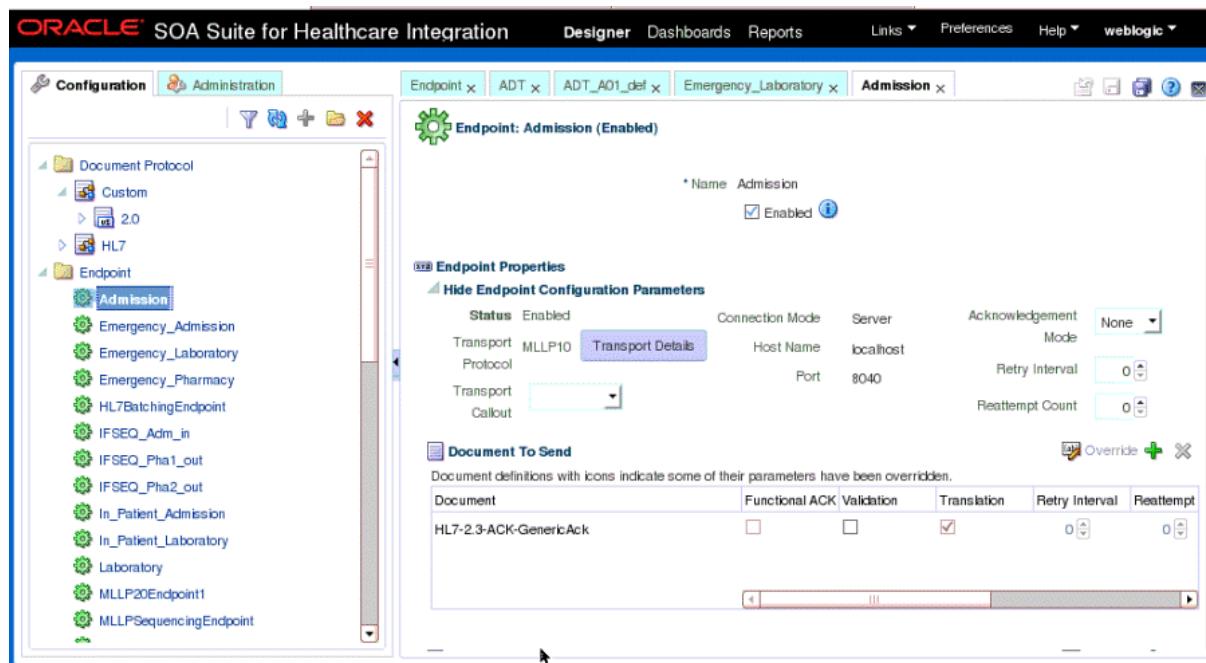
See Working with Endpoints for more information on endpoints.

1.4.1.3 Configuring Callouts

Callouts are used for customizing message processing. For example, callouts can be used to update a message or convert the message format of a remote endpoint to another format. You can use the **Callout** folder in the **Configuration** tab to create, modify, or delete callouts.

Figure 1-5 displays a sample callout.

Figure 1-5   Sample Callout
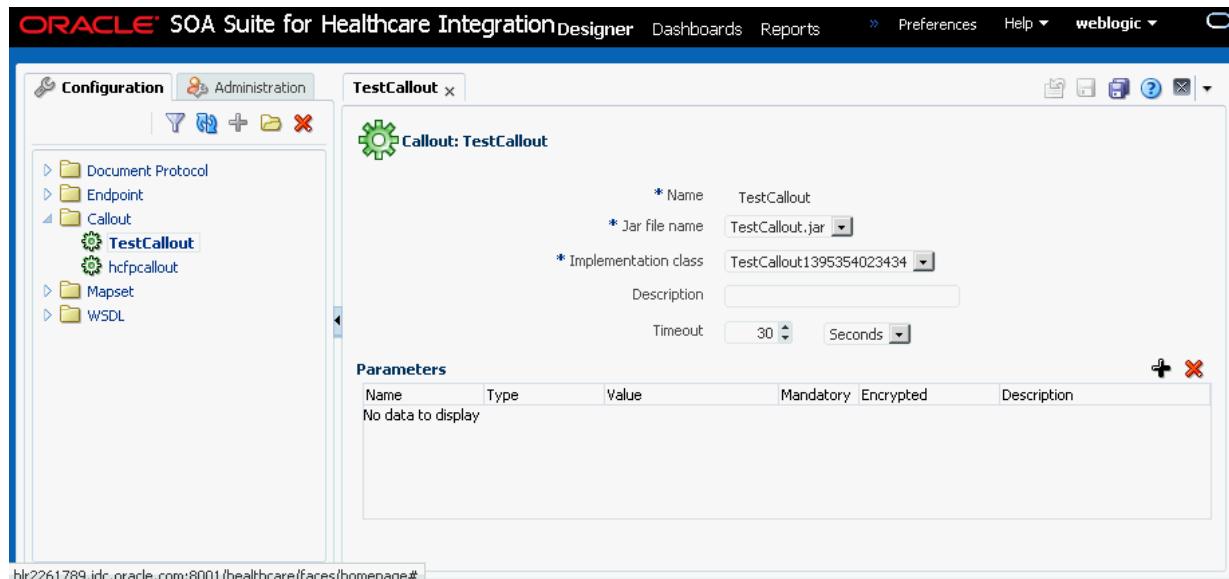
See Working with Callouts for more information on callouts.

1.4.1.4 Configuring Mapsets

Mapsets enable you to directly transform one native data format to another native format. You can use the **Mapset** folder in the **Configuration** tab to create, modify, or delete mapsets.

Figure 1-6 displays a sample mapset.

Figure 1-6   Sample Mapset
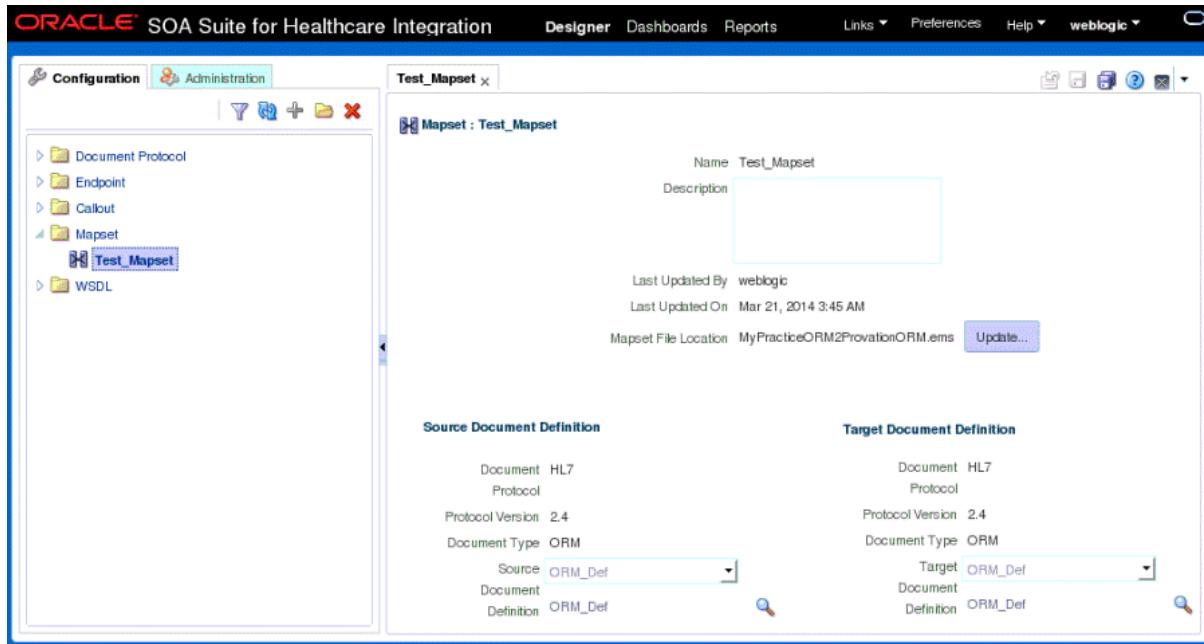
See Working with Mapsets for more information on mapsets.

1.4.1.5 Configuring Internal Delivery Channels

An internal delivery channel defines how a message is delivered from Oracle SOA Suite for healthcare integration to a JMS topic or queue to make it available to internal healthcare systems, or how a message that was sent to a topic or queue from an external system is delivered to Oracle SOA Suite for healthcare integration. It defines the connection information, the transport protocol, and so on. You can use the **Internal Delivery Channel** on the **Administration** tab to create and configure internal delivery channels.

Figure 1-7 displays a sample internal delivery channel.

Figure 1-7   Sample Internal Delivery Channel
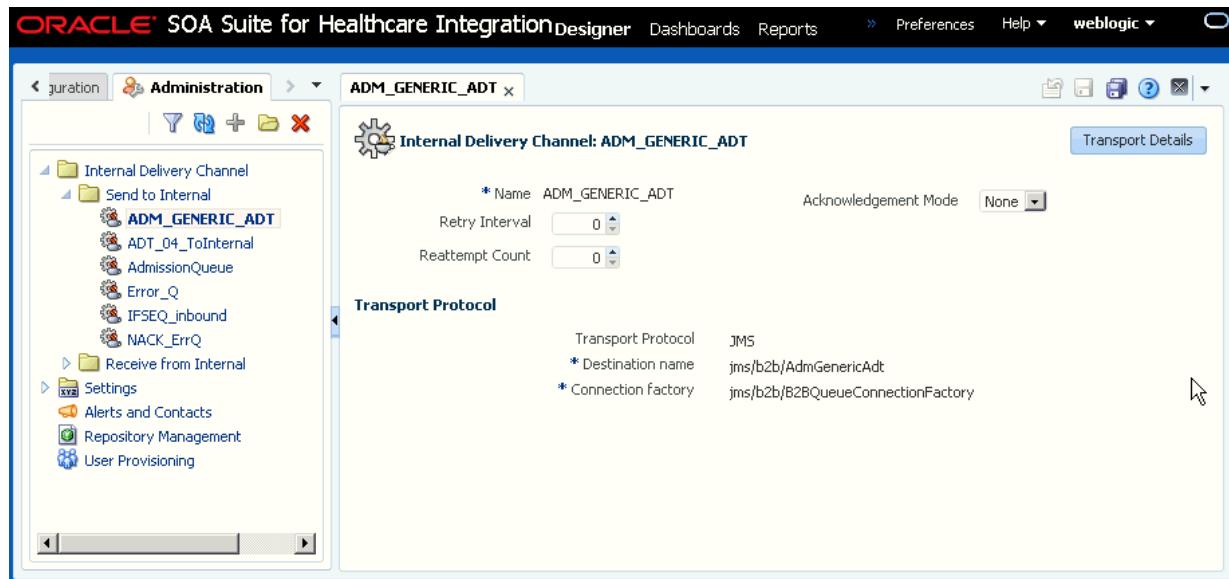
See Working with Internal Delivery Channels for more information on internal delivery channels.

1.4.1.6 Setting runtime and User Interface Properties

Oracle SOA Suite for healthcare integration enables you to modify runtime and user interface properties. Use the **Runtime** and **UI** icons available under the **Settings** folder in the **Administration** tab for modifying these properties:

- runtime settings: Double-click the **Runtime** button to display the **Runtime Settings** page, where you can modify runtime parameters related to functional acknowledgment, large payloads, callout directory names, and dispatcher counts.

  Figure 1-8 displays the Runtime Settings page.

Figure 1-8   The Runtime Settings Page
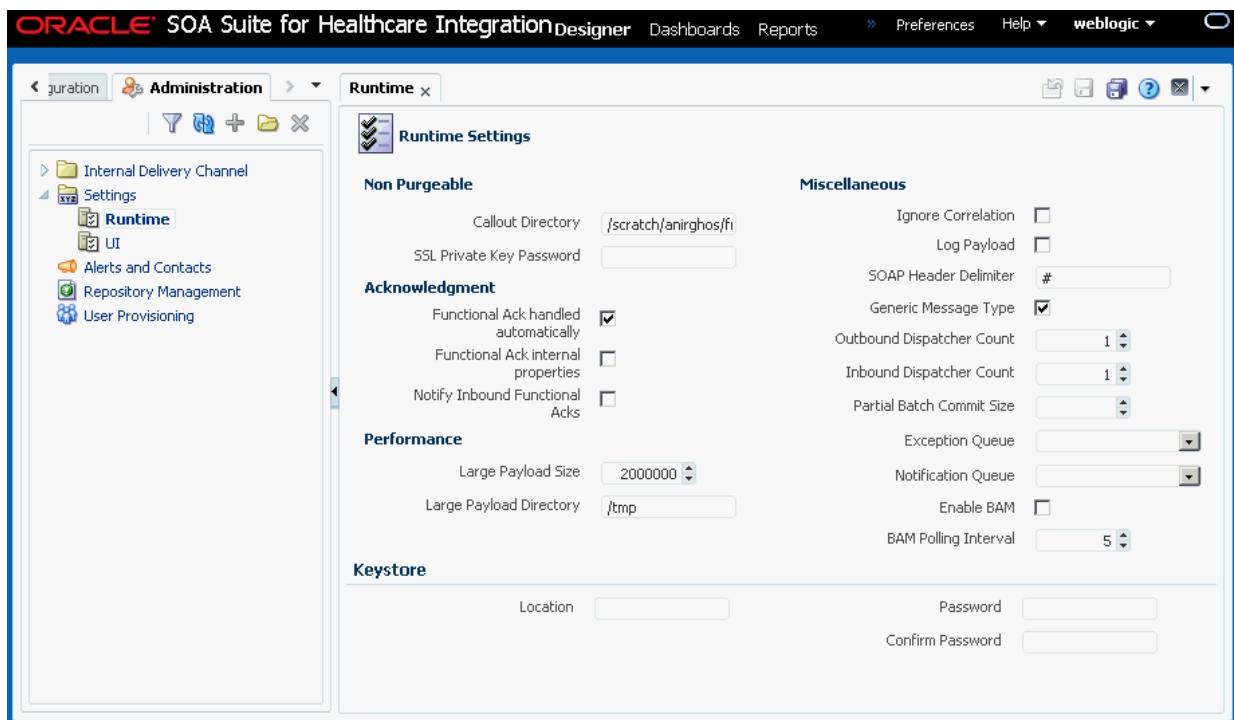
- User Interface settings: Double-click the **UI** button to display the **User Interface Settings** page where you can specify the following:

  - Time slider periods: Enables you to select up to nine time slider periods to be made available for Time Range selection in the Dashboard Summary page

  - Dashboard Summary settings: Enables you to specify historical time range to show statistics for Dashboards, number of columns to use in the endpoint status grid, and auto-refresh interval (in seconds) for the Dashboard Summary view

  - Endpoint settings: Enables you to specify display style for Document Type statistics (Gauge or Table), number of columns to use in Gauges grid, and auto-refresh interval (in seconds) for the Endpoint Detailed view of the Dashboard

  - Reports settings: Enables you to specify auto-refresh interval (in seconds) for reports, number of rows per page to be shown in the Reports Result table, and display of message payload

  Figure 1-9 displays a sample User Interface Settings page.

Figure 1-9   The User Interface Settings Page
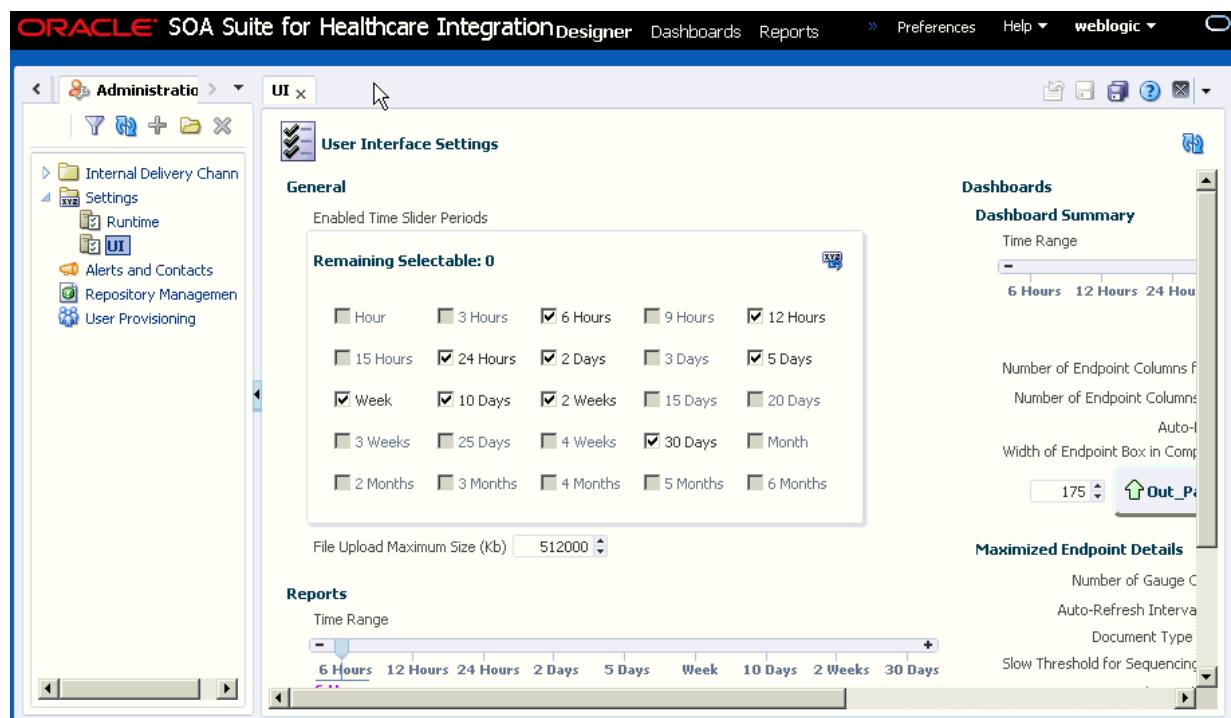
See Configuring System Settings for more information on runtime and user-interface settings.

1.4.1.7 Managing Repository Data

The Repository Management page allows you to export Oracle SOA Suite for healthcare integration repository data and to import other exported files, such as document definitions, mapsets, or other healthcare-related repository data. You can also purge design-time and runtime metadata.

Figure 1-10   Repository Management Page
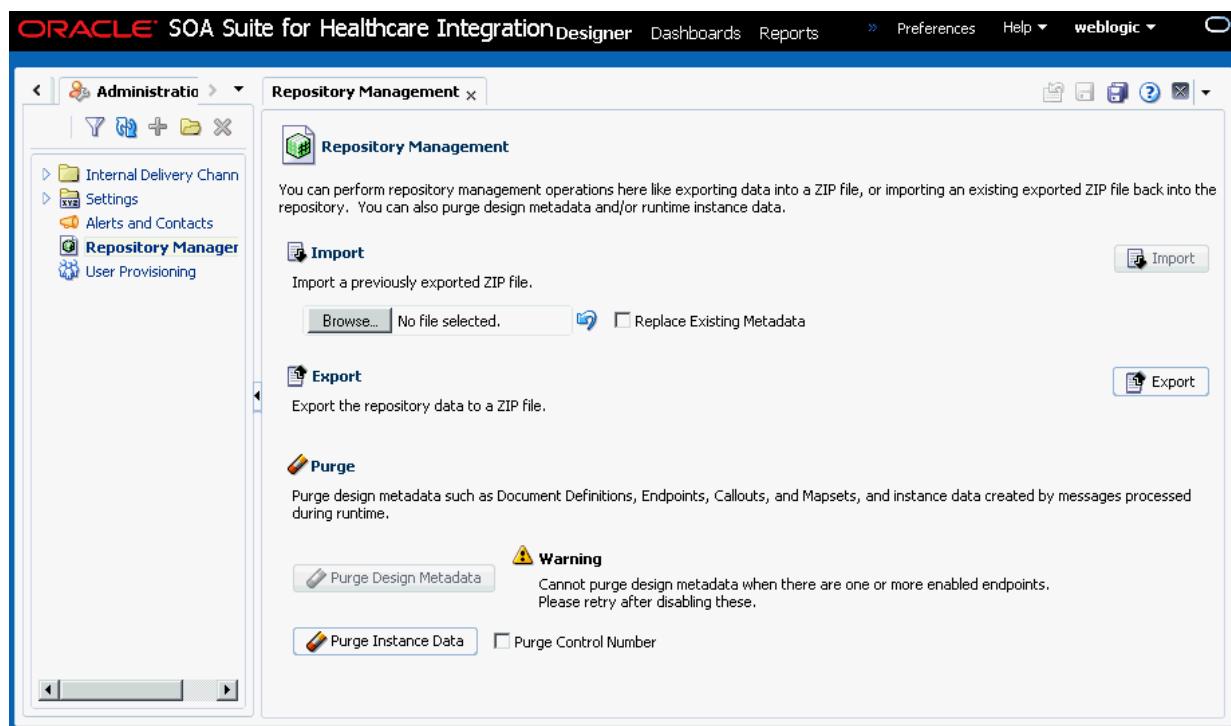
See Managing the Repository for more information on managing the Oracle SOA Suite for healthcare integration repository.

1.4.1.8 Configuring Alerts and Contacts

The Alerts and Contacts page allows you to define a list of contacts and then associate each contact with the events of which they should be notified. An alert can be sent for a variety of runtime and design-time events.

Figure 1-11   Alerts and Contacts Page
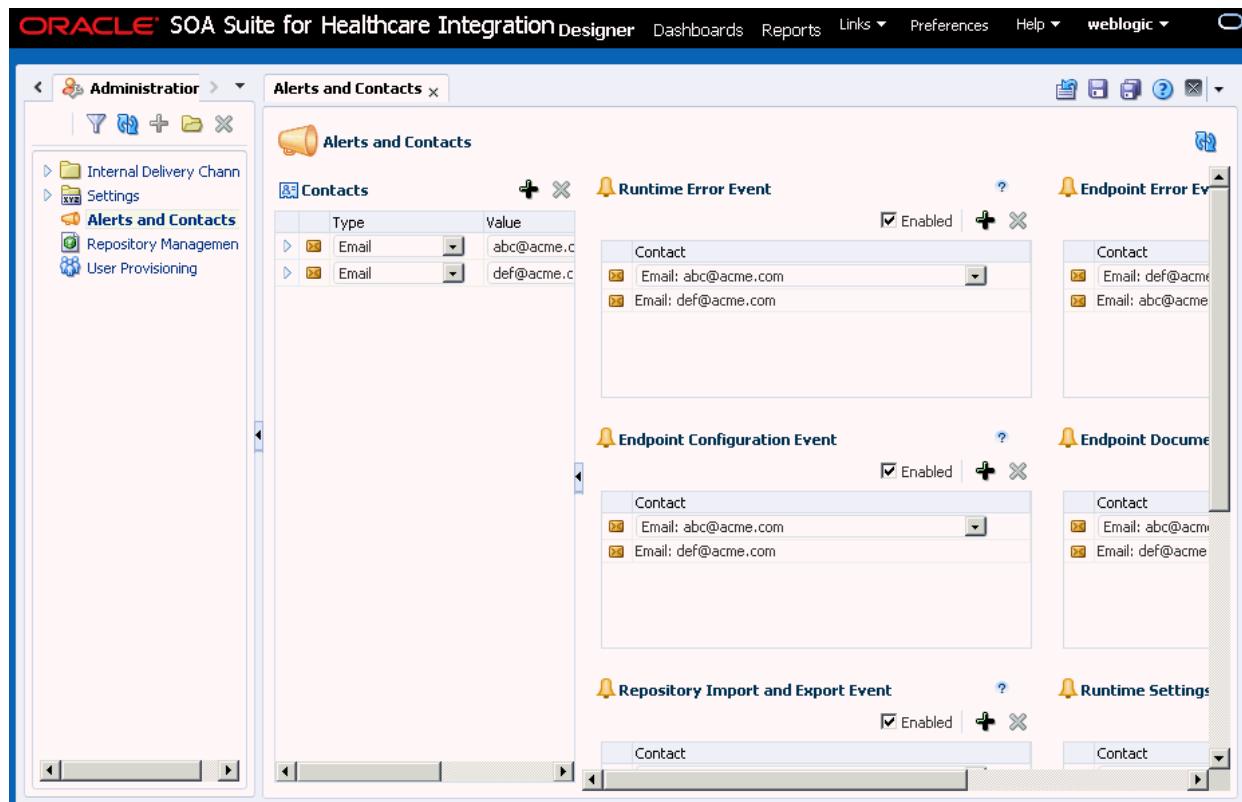
See Configuring Alerts and Contacts for more information on managing alerts and contacts.

1.4.2 Dashboards

Clicking the **Dashboards** tab displays the Dashboards page. When you open the Dashboards page for the first time, the following options are displayed as shown in Figure 1-12:

- **Set Up Default Dashboard to Show Whenever Tab Is Selected**: Specify the dashboard name that is displayed as a default when you open the Dashboards page

- **Open or Edit Existing Dashboards**: Open or edit available dashboards

- **Create Dashboard**: Create a new dashboard

- **Delete Existing Dashboards**: Delete a dashboard

Figure 1-12   The Dashboard Page
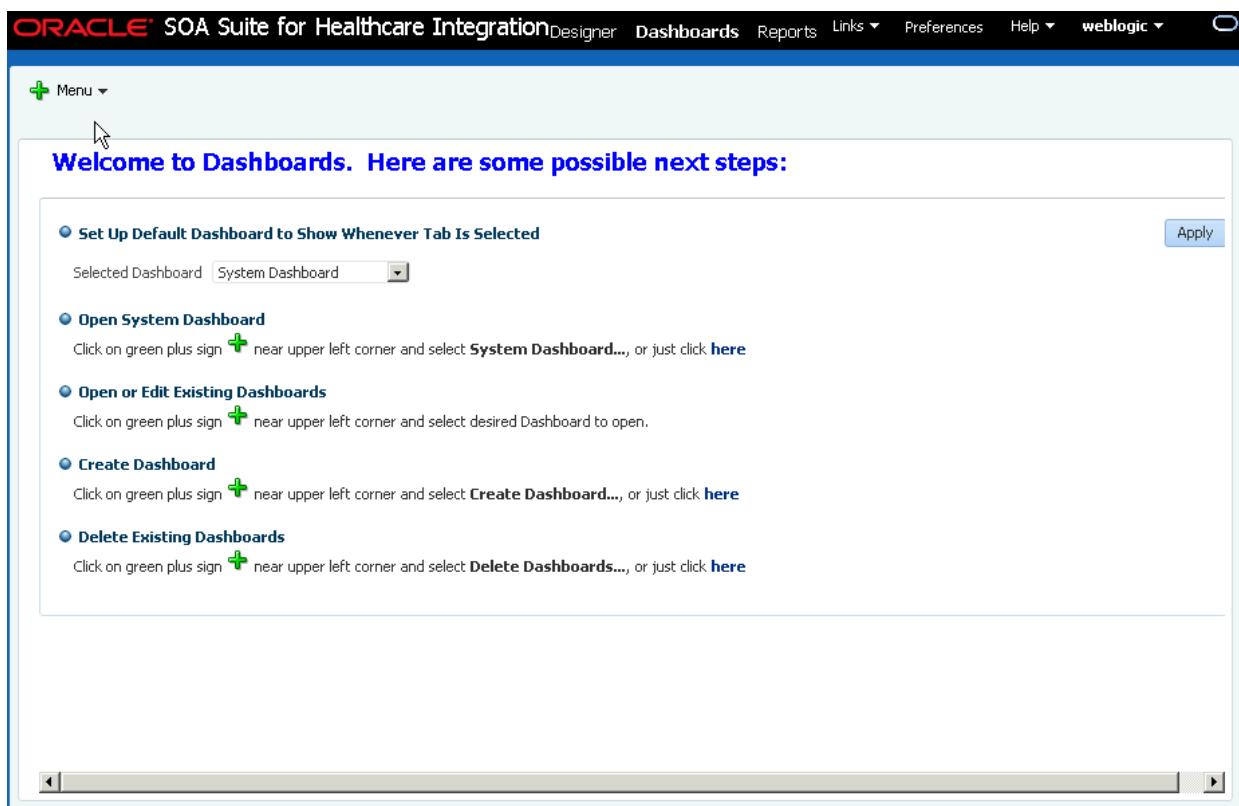
Subsequent access to the Dashboards page displays the default dashboard page that you have specified earlier. On this page, you can see all the endpoints associated with a particular dashboard. In addition, the page enables you to specify whether to show all endpoints regardless of their state, number of columns in which the endpoints are displayed, and auto-refresh interval for the dashboard. The page also provides a **Dashboard Editor** button that enables you to modify the dashboard name, as well as select the required endpoints for the dashboard from a list of available endpoints.

Each endpoint provides options that enable you to view the endpoint properties and also to configure the endpoint. Moving the mouse cursor over the information button of a particular endpoint displays the endpoint properties. When you click the **Configure This Endpoint** button, which is located next to the information button, you are redirected to the relevant endpoint edit page.

You can click the button on upper right corner of the endpoint window to view the details.

Figure 1-13 displays a sample Dashboard page.

Figure 1-13   Sample Dashboard
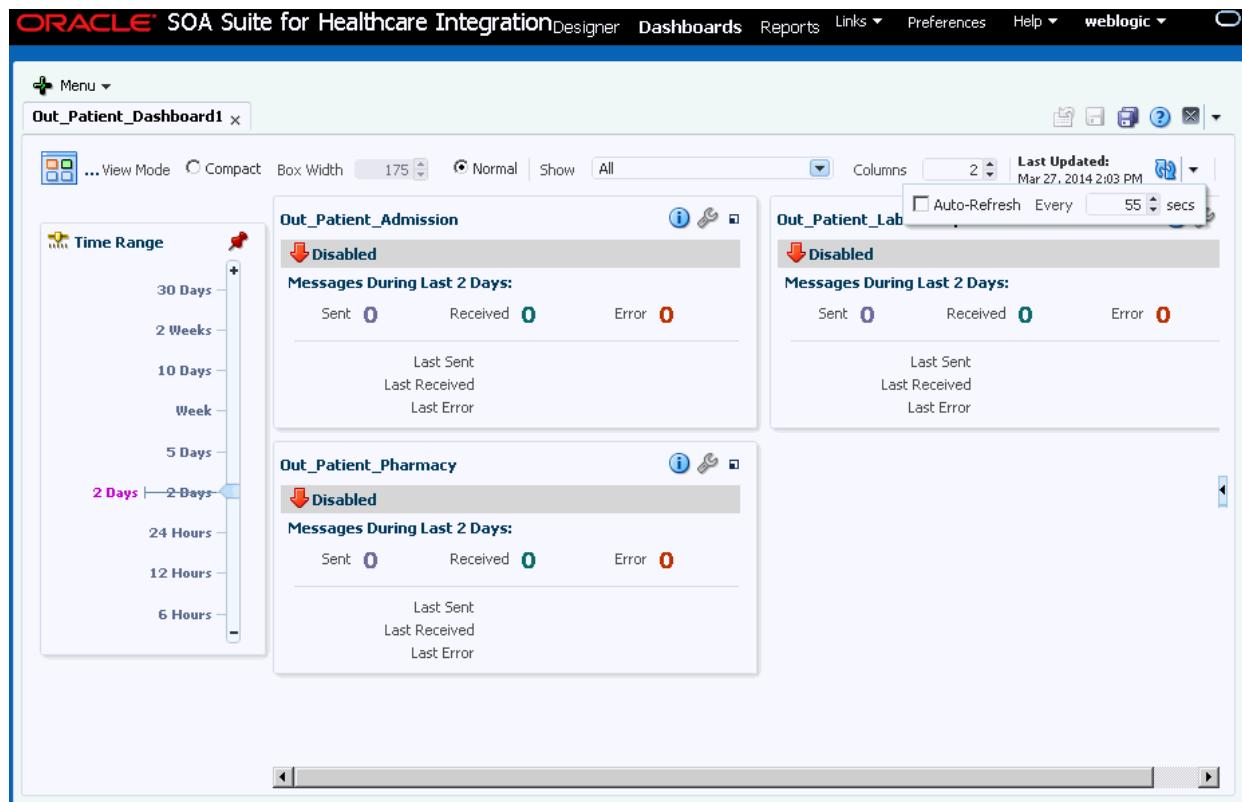
See Working with Dashboards for more information on dashboards.

1.4.3 Reports

Clicking the **Reports** tab displays the Reports page. The Reports page lists all the different types of inbound and outbound messages (wire, business, and application) pertaining to all the endpoints. In addition, the page enables you to:

- Navigate through the available messages

- Specify the number of messages to be displayed per page

- Open a message filter editor to create business message report filters

- Specify auto-refresh interval

When you click a message from the list the lower pane displays all the different message types for the selected message, which are Wire Message, Business Message, and Application Message. Clicking each of the message type displays the details.

Figure 1-14 displays a sample report containing a list of the messages, message type details, and the Business Message Report Editor.

Figure 1-14   Sample Report

See Working with Reports for more information on reports.

1.5 Accessibility Options

This section describes accessibility options available with Oracle SOA Suite for healthcare integration.

1.5.1 Enabling Accessibility Features in Oracle SOA Suite for Healthcare Integration

Oracle SOA Suite for healthcare integration provides the screen reader option, that enables your screen reader to access and read all components of the application.

To enable the screen reader:

1. Click the **Preferences** link in the top right corner of the main window.

2. In the Preferences popup, select the **Accessibility** option in the navigation pane.

3. In the **Mode Settings** dropdown list, select **Enable Screen Reader Mode**.

4. Click **OK**.

2

Working with the Oracle Healthcare Adapter

This chapter describes the Oracle Healthcare Adapter, which enables healthcare integration components to be used in Oracle SOA Suite composites. The adapter is used in an Oracle JDeveloper environment.

This chapter includes the following topics:

- Introduction to the Oracle Healthcare Adapter
- How to Use Healthcare Adapters in a SOA Composite Application

2.1 Introduction to the Oracle Healthcare Adapter

The Oracle Healthcare Adapter allows you to add healthcare integration binding components to a SOA composite application to create an end-to-end process, such as sending admissions information generated by a registration application to a laboratory system.

The Healthcare Adapter establishes the type of documents transmitted between a SOA composite application and external healthcare applications. You can use other SOA Suite components in your composite application, including BPEL processes, Oracle Mediator components, a variety of adapters, and so on.

2.1.1 The Healthcare Configuration Wizard

Adding a Healthcare Adapter to a composite launches the Healthcare Configuration Wizard in Oracle JDeveloper. The wizard lets you create and configure healthcare integration binding components in a SOA composite application as follows:

- The component is used as a *service* (inbound) to receive messages from external systems and deliver them to SOA composite applications. Oracle SOA Suite for healthcare integration is the entry point to the SOA composite application.

- The component is used as a *reference* (outbound) to send messages from the SOA composite application to external applications.

As you follow the steps in the Healthcare Configuration Wizard, you are prompted to select components, such as document definitions, which were created in Oracle SOA Suite for healthcare integration. You can also launch Oracle SOA Suite for healthcare integration from the wizard to create a document definition if the right one does not already exist.

> **Note:**
>
> If SSL is enabled in the middleware (the healthcare integration Web service), then the Healthcare Configuration Wizard detects the SSL port and retrieves the document definitions using the SSL connection.

2.1.2 What Happens When You Add a Healthcare Adapter to a SOA Composite

Completing the Healthcare Configuration Wizard adds a new healthcare integration service or reference to the composite application and generates the corresponding WSDL file. The WSDL file defines the schema, message, document definition, the WebLogic Managed Server, and the connection to the application server. Connecting the service or reference to Oracle BPEL processes or Oracle Mediator component makes the healthcare integration normalized message properties available to that process or component so you can further configure the healthcare application by specifying endpoints, message types, message IDs, document types, protocols, and so on.

> **Note:**
>
> In order to connect the composite application to a healthcare integration endpoint, you must specify a value for the **hc.fromEndpoint** normalized message property if a healthcare adapter is defined as a service, and you must specify a value for the **hc.toEndpoint** normalized message property if a healthcare adapter is defined as a reference.

2.2 How to Use Healthcare Adapters in a SOA Composite Application

To create a SOA composite application with a Healthcare Adapter, perform the following steps in the order given:

- Create and enable the healthcare integration endpoints (described in Working with Endpoints.)

- Creating a SOA Application and Project

- Add Service Components

- Adding a Healthcare Integration Binding Component

2.2.1 Creating a SOA Application and Project

You can use the Healthcare Adapter to add a healthcare integration binding component to an existing JDeveloper project to enable communications between the SOA Suite and external healthcare systems. First you must create the application and the project.

For more information about creating SOA composite applications, see *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite*.

Before You Begin

Before you create the application and project, review the following guidelines:

- Do *not* use spaces in the name of the application.

- Do *not* create applications and projects in directory paths that have spaces (for example, `c:\Program Files`).

- In a UNIX operating system, it is highly recommended that you enable Unicode support by setting the `LANG` and `LC_All` environment variables to a locale with the

UTF-8 character set. This enables the operating system to process any character in Unicode. SOA technologies are based on Unicode. If the operating system is configured to use non-UTF-8 encoding, SOA components might function in an unexpected way.

To enable Unicode support in an Oracle JDeveloper design-time environment, select **Tools** > **Preferences** > **Environment** > **Encoding** > **UTF-8**. This setting is also applicable for runtime environments.

- Composite and component names cannot exceed 500 characters.

- A project deployed to the same infrastructure *must* have a unique name across all SOA composite applications. The uniqueness of a composite is determined by its project name. For example, do *not* create a project named **Project1** in two different applications. During deployment, the second deployed project (composite) overwrites the first deployed project (composite).

To create a SOA application and project

1. Start Oracle JDeveloper Studio Edition.

   > **Note:**
   >
   > If you are starting Oracle JDeveloper for the first time, specify the location for the Java JDK.

2. Create a new SOA composite application, as described in Table 2-1.

   Table 2-1   SOA Composite Application Creation

If Oracle JDeveloper...	Then...
Has no applications For example, you are opening Oracle JDeveloper for the first time.	In the Application Navigator in the upper left, click **New Application**.
Has existing applications	From the **File** main menu or the **Application** menu: **a.** Select **New** > **Applications**. The New Gallery opens, where you can select different application components to create. **b.** In the **Categories** tree, under the **General** node, select **Applications**. In the **Items** pane, select **SOA Application** and click **OK**.

   The Create SOA Application wizard appears.

3. On the Name your application page, you can optionally change the name and location for your project. If this is your first application, from **Application Template**, select **SOA Application**. Accept the defaults for the package prefix, and click **Next**.

4. On the Name your project page, you can optionally change the name and location for your SOA project. By default, Oracle JDeveloper adds the SOA project technology, the `composite.xml` that generates, and the necessary libraries to your model project. Click **Next**.

   The Project SOA Settings page of the Create SOA Application wizard appears.

5. In the Configure SOA Settings page, click **Empty Composite**, and click **Finish**.

6. Select **Save All** from the **File** main menu.

2.2.2 Adding a Healthcare Integration Binding Component

You can add a healthcare integration binding component as an exposed service (inbound) or an external reference (outbound) to define the connection to a healthcare system.

Oracle Healthcare supports the following integrations with the Healthcare adapter:

- Adding a Default (Fabric) Integration Binding Component

- Adding a JMS Integration Binding Component

2.2.2.1 Adding a Default (Fabric) Integration Binding Component

You can add a Fabric integration binding component by using the Healthcare Configuration Wizard.

Before you begin:

Before you add a healthcare integration binding component to an Oracle SOA project, make sure you have created the necessary components in the healthcare integration user interface. For example, make sure to create and configure any document definitions and endpoints that you use in the project.

To add a fabric healthcare integration binding component

1. From the Component Palette, select **SOA**.

2. Scroll to **Service Adapters** and drag a **Healthcare Adapter** to either the **Exposed Services** or **External References** swim lane.

   - Drag the adapter to **Exposed Services** for receiving inbound messages.

   - Drag the adapter to **External References** for sending outbound messages.

   The Healthcare Configuration Wizard appears.

3. On the Healthcare Configuration Wizard Welcome page, click **Next**.

   The Service Name page appears.

4. Enter a name for the healthcare service and click **Next** as shown in Figure 2-1.

Figure 2-1   Healthcare Configuration Wizard - Service Name Page

The Healthcare Integration Type page appears.

5. Select **Default** and click **Next** as shown in Figure 2-2.

Figure 2-2   Healthcare Configuration Wizard - Healthcare Integration Type Page

The Application Server Connection page appears.

6. Do one of the following:

    - From the **AppServer Connection** list, select an application server connection and click **Next**.

    - Click **New** to create an application server connection. Follow the Create Application Server Connection Wizard.

When the connection is established, the user name and host name appear along with the name of the SOA Server as shown in Figure 2-3.

Figure 2-3   Healthcare Configuration Wizard - Application Server Connection Page
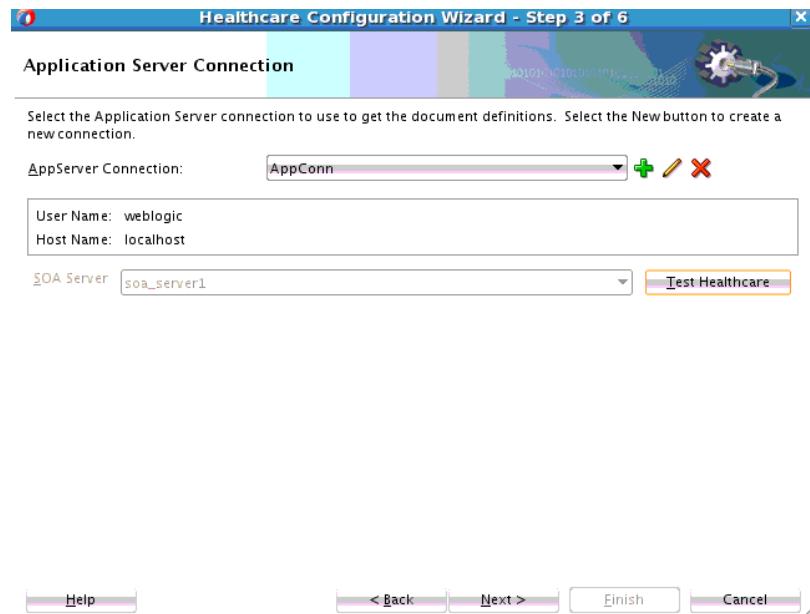
7. To verify the connection to the application server, click **Test Healthcare**, and then click **OK** on the dialog that appears.

8. Click **Next**.

The Operation page appears as shown in Figure 2-4.

Figure 2-4   Healthcare Configuration Wizard - Operation Page

9. Select **Send** for outbound messages or select **Receive** for inbound messages.

10. Click **Next**.

The Document Definition Handling page appears.

11. Retain the default (to import the schema from Oracle SOA Suite for healthcare integration) or select one of the options on the **Advanced** tab, as described in Table 2-3 and shown in Figure 2-5.

Figure 2-5 Healthcare Configuration Wizard - Document Definition Handling Page
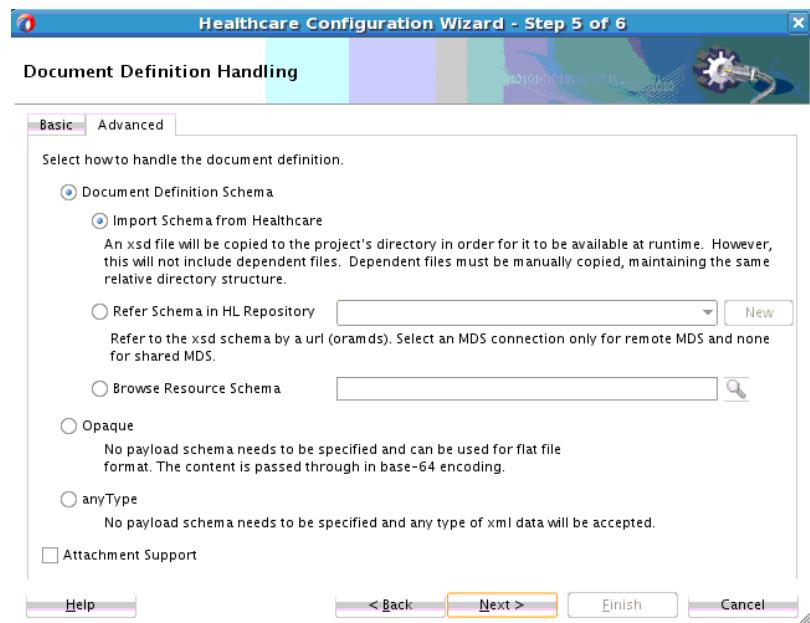
12. Click **Next**.

The Document Definition page appears.

13. Expand the tree to select a document definition as shown in Figure 2-6.

Figure 2-6   Healthcare Configuration Wizard - Document Definition Page
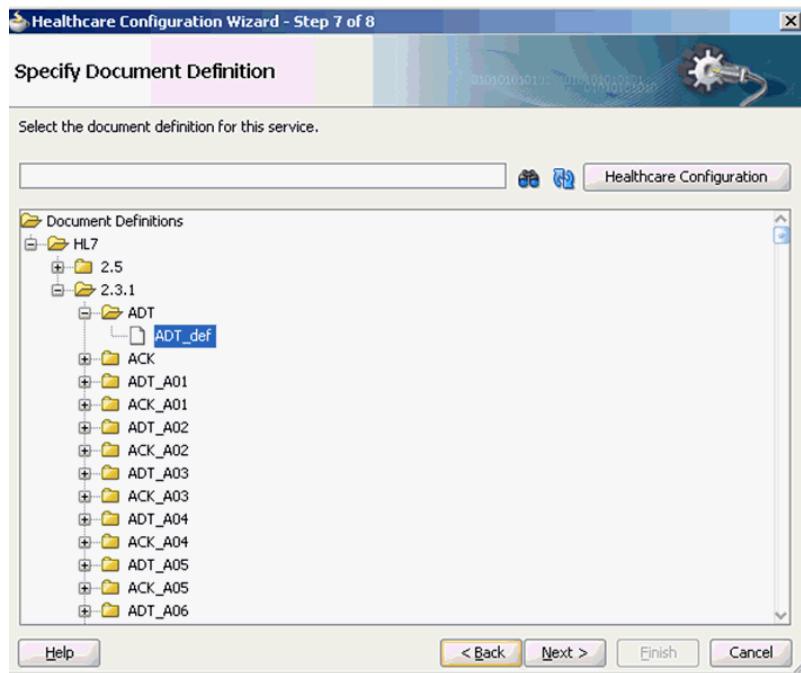
Table 2-2 describes additional options on the page.

Table 2-2   Document Definition Page Options

Option	Description
Search	Enter a definition name in the empty field and click the Search button. Partial strings are matched if you type the beginning of the definition name. Partial strings with wildcards cannot be used.
Refresh	Retrieves the document definition list from the healthcare integration server. Refresh after a search to see all document definitions.
Healthcare Configuration	Opens a browser to the Oracle SOA Suite for healthcare integration user interface, using the connection specified on the Application Server Connection page. In the healthcare integration user interface, you can create or import a document definition. After you finish those steps, return to this dialog, click **Refresh**, and select the new document definition.

14. Click **Next**.

15. If you selected a document definition with multiple root elements, the Root Elements page appears. Select a root element to use and click **OK**.

16. Click **Next** and then click **Finish**.

2.2.2.2 Adding a JMS Integration Binding Component

You can add a Fabric integration binding component by using the Healthcare Configuration Wizard.

Before you begin:

Before you add a healthcare integration binding component to an Oracle SOA project, make sure you have created the necessary components in the healthcare integration user interface. For example, make sure to create and configure any document definitions and endpoints that you use in the project.

To add a JMS healthcare integration binding component

1. Repeat steps 1-3 from Adding a Default (Fabric) Integration Binding Component..

2. Scroll to **Service Adapters** and drag a **Healthcare Adapter** to either the **Exposed Services** or **External References** swim lane.

   • Drag the adapter to **Exposed Services** for receiving inbound messages.

   • Drag the adapter to **External References** for sending outbound messages.

   The Healthcare Configuration Wizard appears.

3. On the Healthcare Configuration Wizard Welcome page, click **Next**.

   The Service Name page appears.

4. Enter a name for the healthcare service (say hcfp_Inbound to receive inbound messages) and click **Next** as shown in Figure 2-7.

   Figure 2-7   Healthcare Configuration Wizard - Service Name Page

   The Healthcare Integration Type page appears.

5. Select **JMS** and click **Next** as shown in Figure 2-2.

Figure 2-8   Healthcare Configuration Wizard - Healthcare Integration Type Page
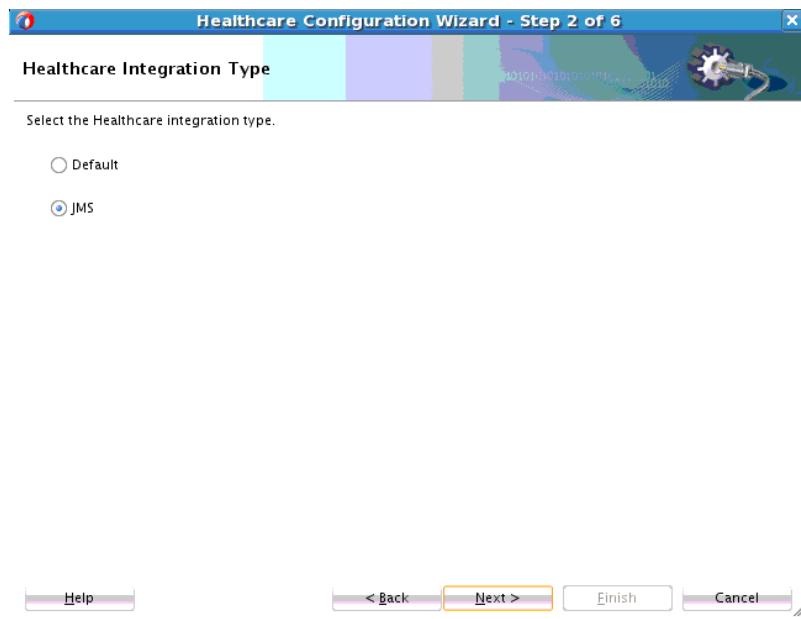
The Application Server Connection page appears.

6. Repeat steps 6-8 from Adding a Default (Fabric) Integration Binding Component.

7. In the Operations page, select **Send** for outbound messages or select **Receive** for inbound messages. In this case select **Receive** as shown in Figure 2-9.

Figure 2-9   Healthcare Configuration Wizard - Operation Page

8. Click **Next**.

The Document Definition Handling page appears.

9. Repeat steps 11-16 from Adding a Default (Fabric) Integration Binding Component.

10. In the JMS Provider page, select **Oracle Weblogic JMS** from the Oracle Enterprise Messaging Service (OEMS) list and click **Next** as shown in Figure 2-10.

Figure 2-10   Healthcare Configuration Wizard - JMS Provider Page
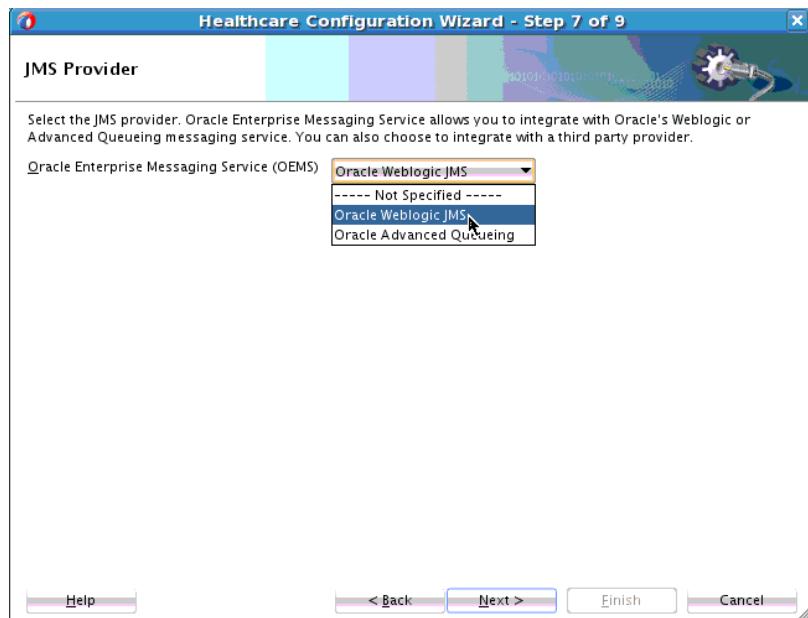
The Service Connection page appears.

11. Select the Application Server connection from the AppServer Connection list (or create a connection by clicking the plus button and then using the Create Application Server Connection wizard) and click **Next**.

The Consume Operation Parameters page appears.

12. Click the **Browse** button to display the Select Destination dialog box.

13. Select the required destination name from the list. In this case, select **jms/b2b/B2B_IN_QUEUE** (to receive inbound messages) and click **OK** to go back to the Consume Operation Parameters page. Note that the JNDI Name field is already populated.

14. Click **Next** as shown in Figure 2-11.

Figure 2-11   Healthcare Configuration Wizard - Consume Operations Page
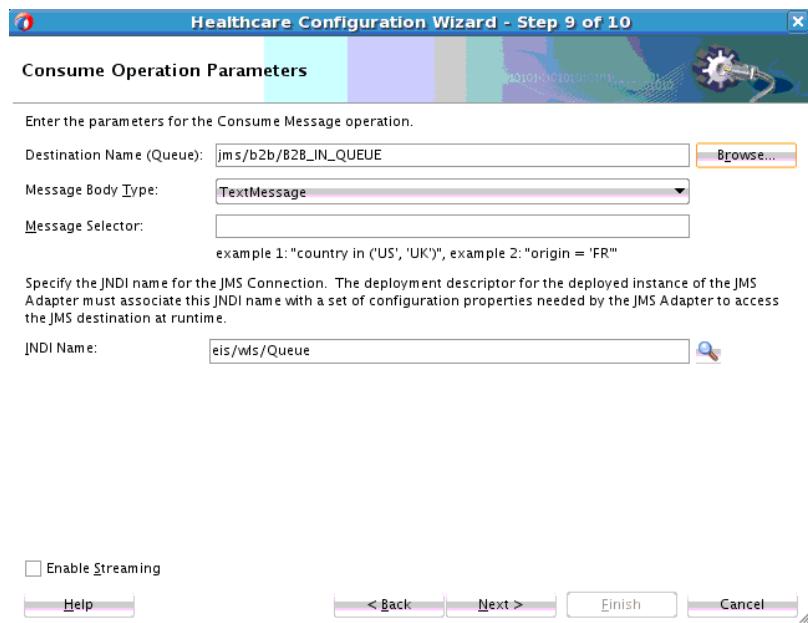
Note:

When you select **Receive** in the Operation page, the Consume operation is pre-selected and the Consume Operation Parameters page is displayed.

Likewise, when you select **Send** in the Operation page, the Produce operation is pre-selected and the Produce Operation Parameters page is displayed. For a Send operation, you must select the destination of the B2B_OUT_QUEUE as shown in the following figure.
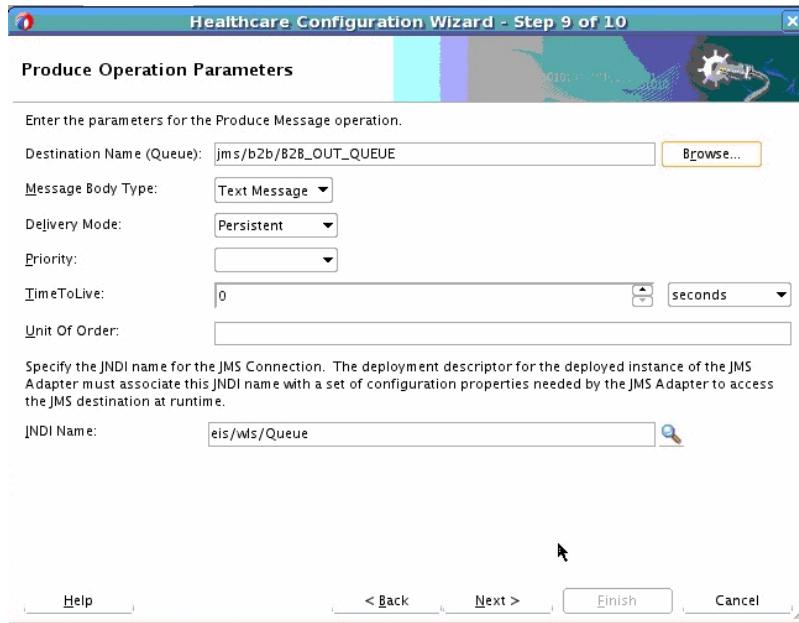
15. Click **Next** and then click **Finish**.

2.2.2.3 Document Definition Handling in the Healthcare Configuration Wizard

The Healthcare Configuration Wizard lets you associate document definitions from Oracle SOA Suite for healthcare integration with the adapter you are creating. You can specify how the document definition is associated with the Composite and whether the schema definition (XSD) file is required for validation. Table 2-3 lists and describes the possible options for handling the document definition.

Table 2-3   Advanced Document Definition Handling Options

Option	Description
Import Schema from Healthcare	Imports the schema from Oracle SOA Suite for healthcare integration. This option copies the XSD file to the project directory to make it available at runtime. If there are any dependent files, you must copy them manually to the project, maintaining the same directory structure.
Refer Schema in HL Repository	Uses an existing metadata service (MDS) connection or allows you to create a new connection to use. Select an existing service or create a new MDS connection. If you create a new MDS connection, the MDS Connection Wizard appears so you can define a connection. This connection is required to access the Oracle SOA Suite for healthcare integration repository. When you select a document definition, a URL is generated to link to the MDS.
	The selected application server connection refers to a specific Oracle SOA Suite for healthcare integration instance. The MDS connection used by the instance must match the selected MDS connection to avoid inconsistent document definitions.
	When referring to a schema in an Oracle SOA Suite for healthcare integration repository, an MDS connection is required only for referring to a schema in a remote MDS, but not if the schema is referred to within the local shared MDS repository.
Browse Resource Schema	Browse for a schema using the SOA Resource Browser. Selecting this option and clicking the Browse Schema button opens the Type Chooser dialog. Expand the tree, select a type, and return to the Document Definition Handling page.
Opaque	Handles any type of data (for example, positional flat file) when the content is passed through in base-64 encoding. You need not specify a schema.
anyType	Handles any type of XML data. You need not specify a schema.

2.2.3 Add Service Components

After you create the SOA application and project, you can add the service components that implement the business logic and processing rules for the application.

To add service components to a project

1. From the Component Palette, select **SOA**.

2. From the **Service Components** list, drag a component into the designer.

Figure 2-12 shows a BPEL process just added to the designer.

Figure 2-12 BPEL Process in a Composite Application
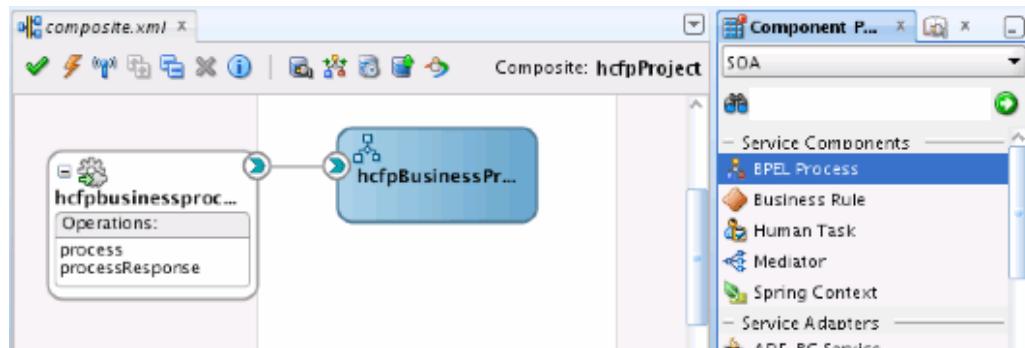
A dialog specific to the selected service component appears. Table 2-4 describes the available dialogs.

Table 2-4 Starting Service Component Editors

Dragging This Service Component...	Invokes The...
BPEL Process	Create BPEL Process dialog to create a BPEL process that integrates a series of business activities and services into an end-to-end process flow.
Business Rule	Create Business Rules dialog to create a business decision based on rules.
Human Task	Create Human Task dialog to create a workflow that describes the tasks for users or groups to perform as part of an end-to-end business process flow.
Mediator	Create Mediator dialog to define services that perform message and event routing, filtering, and transformations.

3. Configure the settings for the service component. For help with a service component dialog, click **Help** or press **F1**.

4. When configuring the component, define any of the normalized message properties listed in Table 2-5. **hc.fromEndpoint** and **hc.toEndpoint** must be defined and must correspond to endpoints defined in the healthcare integration user interface.

Table 2-5 Healthcare Integration Normalized Message Properties

hc.documentDefinitionName	The name of the document definition. For example, **ADT_A03_def**.
hc.documentProtocolName	The name of the protocol associated with the document definition. For example, **HL7**.

Table 2-5   (Cont.) Healthcare Integration Normalized Message Properties

hc.documentProtocolVersion	The version of the above protocol associated with the document definition. For example, **2.3.1** for an HL7 document.
hc.documentTypeName	The document type associated with the document definition. For example, **ADT_A03** for an HL7 document.
hc.fromEndpoint	The name of the endpoint for the sending application. This name is defined in the endpoint configuration in the healthcare integration user interface.
hc.messageId	A unique message ID.
hc.messageType	The message type value can be one of the following options: • 1 (indicates a request) • 2 (indicates a response) • 9 (indicates a functional acknowledgment)
hc.replyToMessageId	The message ID to which the sending message replies.
hc.toEndpoint	The name of the endpoint for the receiving application. This name is defined in the endpoint configuration in the healthcare integration user interface.

5. When you are done configuring the settings, click **OK**.

6. Select **Save All** from the **File** main menu.

See *Oracle Fusion Middleware Developer's Guide for Oracle SOA Suite* for more information about adding service components.

3

Working with Document Types and Protocols

This chapter provides information about the document protocols and message types supported by Oracle SOA Suite for healthcare integration, and describes how to work with document definitions in the healthcare integration user interface.

This chapter contains the following topics:

- Introduction to Document Protocols
- Using the Custom Document Protocol
- Using the HL7 Document Protocol
- Creating Document Definitions
- Deleting a Document Definition

3.1 Introduction to Document Protocols

Oracle SOA Suite for healthcare integration supports the following document protocols:

- Custom
- HL7 V2.x

Figure 3-1 displays the document protocols supported in Oracle SOA Suite for healthcare integration.

Figure 3-1   Oracle SOA Suite for healthcare integration Document Protocols
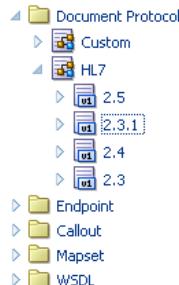
As part of the document definition, you provide the document guideline files, which are typically created in Oracle Document Editor. (For Custom documents, you cannot use Oracle Document Editor and validation of documents is also not possible.) If validation is enabled, then, at runtime, the payload must conform to the document definition file type you use.

3.1.1 What You Might Need to Know About the Document Hierarchy

You can think of a document protocol as a hierarchy, as shown in Figure 3-2.

Figure 3-2   Document Hierarchy

```
Document Protocol
   Document Protocol Version
       Document Type
           Document Definition
```
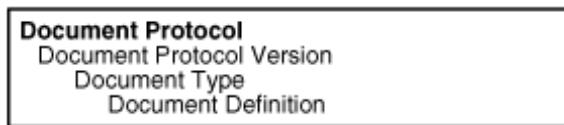
A document protocol can consist of multiple document protocol versions. A document protocol version can consist of multiple document types. A document type can consist of multiple document definitions. Typically, you start with one document definition and customize it for different endpoints.

Figure 3-3 shows a document protocol hierarchy as it applies to HL7 V2.3.1.

Figure 3-3   HL7 V2.3.1 Document Hierarchy
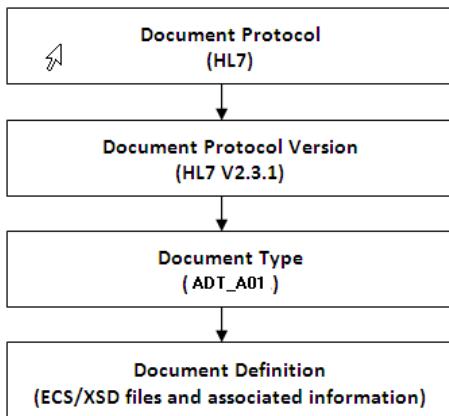
In the Oracle SOA Suite for healthcare integration user interface, as you create a document definition, the document protocol hierarchy is reflected in the definition:

DocumentProtocol—Version—DocumentType—DocumentDefinitionName

Example - Document Definition Name for an HL7 Document shows the hierarchy reflected in the definition for an HL7 document.

Document Definition Name for an HL7 Document

```
Document protocol: HL7Document protocol version: 2.3.1Document type: ADT_A01Document
definition: ADT_A01_defThe resulting document definition is:HL7-2.3.1-ADT_A01-
ADT_A01_def
```

3.1.2 What You Might Need to Know About Document Protocols with Acknowledgments

For any message flow that involves an acknowledgment, Oracle SOA Suite for healthcare integration sends an acknowledgment only once. Resubmission does not generate another acknowledgment if the message has already been acknowledged.

3.2 Using the Custom Document Protocol

Oracle SOA Suite for healthcare integration supports custom document protocols to create documents required for proprietary transactions. With XML messages, you have the advantage of schema enforcement (XSDs).

With non-XML messages, you can create endpoints for specific message types.

When creating a Custom document, you specify rules to identify the incoming document. For XML documents, specify an XPath expression or an XPath expression and a value, which is the expected result of the expression.

For non-XML documents such as a flat file, you can specify Identification Start Position, End Position, and Identification Value.

3.2.1 What You Might Need to Know About Custom Document Version Parameters

No parameters are required to be set when you create the document version for a Custom document.

3.2.2 What You Might Need to Know About Custom Document Type Parameters

When you create a Custom document type, you can set ebXML messaging service (ebMS) parameters to identify the ebXML document. Figure 3-4 shows the document type parameters for a Custom document.

Figure 3-4   Document Type Parameters for a Custom Document
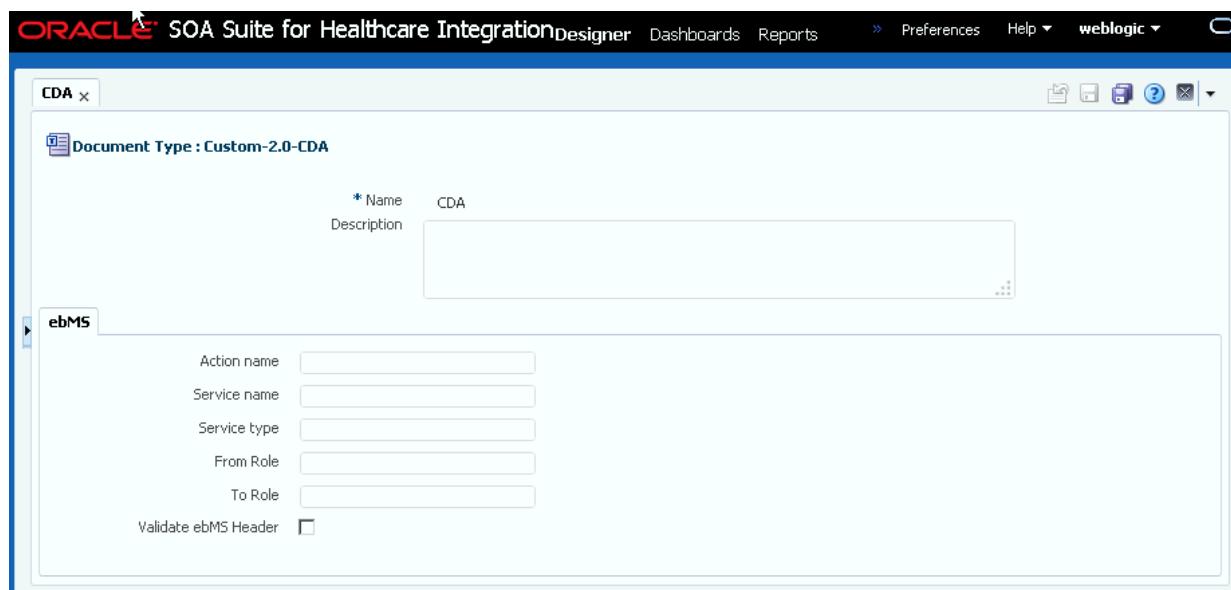
Table 3-1 describes the document type parameters for a Custom document.

Table 3-1   Document Type Parameters for a Custom Document

Parameter	Description
ebMS Tab	-

Table 3-1 (Cont.) Document Type Parameters for a Custom Document

Parameter	Description
Action name	The action name for the ebXML header, which is also an identification criteria for inbound and outbound messages. ebMS documents require an action name to avoid run-time errors.
Service name	The service name for the ebXML header, which is also an identification criteria for inbound messages. ebMS documents require a service name to avoid run-time errors.
Service type	The service type for the ebXML header, which is also an identification criteria for inbound messages. ebMS documents require a service type to avoid run-time errors.
From Role	The endpoint that sends the message. A value provided here overrides the **Identifiers** values supplied on the **Profile** tab.
To Role	The endpoint that receives the message. A value provided here overrides the **Identifiers** values supplied on the **Profile** tab.
Validate ebMS Header	When selected, validates inbound ebMS header from role to role.

3.2.3 What You Might Need to Know About Custom Document Definition Parameters

When you create a Custom document definition (see Creating Document Definitions for more information on creating document definitions), select the identification type —XML or Flat, and set parameters in the tabbed areas. Figure 3-5 shows the document definition parameters for an XML-type Custom document.

Figure 3-5 Document Definition Parameters for an XML-Type Custom Document
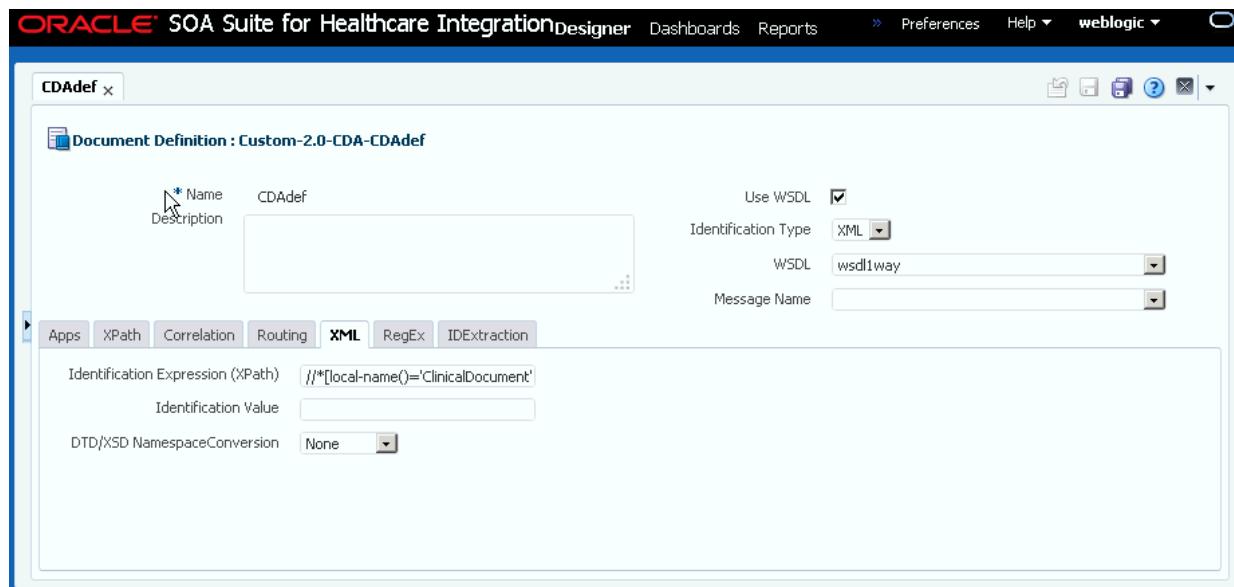
Figure 3-6 shows the document definition parameters for a flat-file Custom document.

Figure 3-6   Document Definition Parameters for a Flat-FIle Custom Document
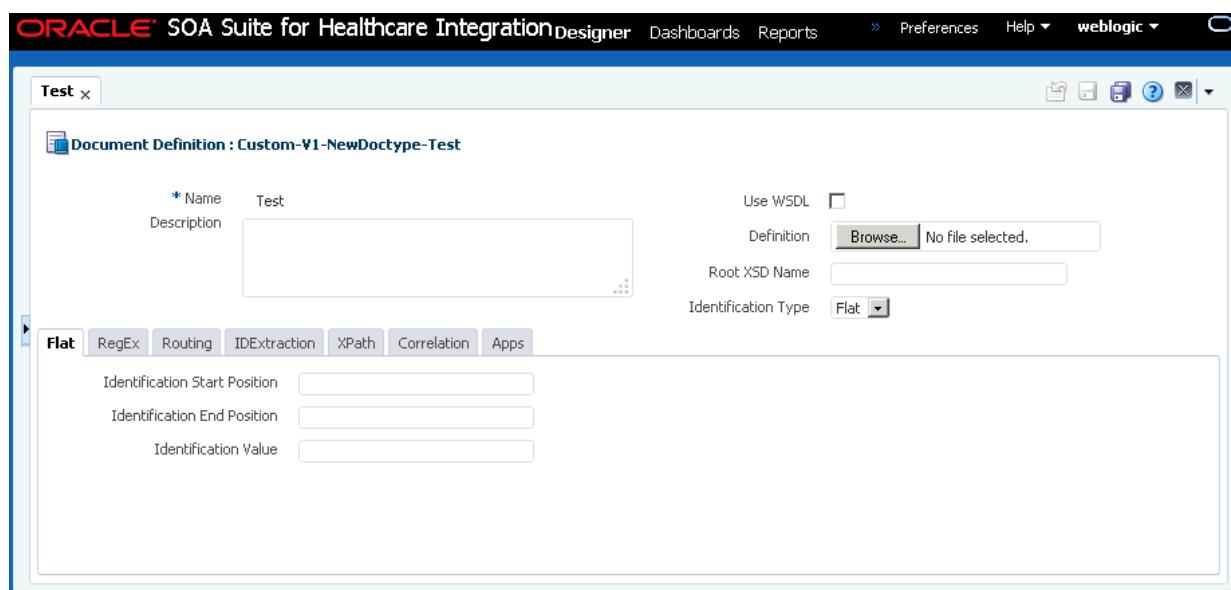
Table 3-2 describes the document definition parameters for a Custom document.

Table 3-2   Document Definition Parameters for a Custom Document

Parameter	Description
XML Tab	(Available if **XML** is selected from **Identification Type**)
Identification Expression (XPath)	Locates a node in the XML payload
Identification Value	Provides the value to match in the node identified by the Identification Expression. If the values match, then the document is successfully identified. If the value is left blank, then Oracle SOA Suite for healthcare integration checks for the existence of the node and the document is successfully identified.
DTD/XSD NamespaceConversion	Select from **None**, **Both**, **Inbound**, or **Outbound**.
Routing Tab	-
Document Routing ID	Sets the consumer name to the back-end application
XPath Tab	See How to Configure the XPath Expression for a Custom XML Document.
XPath Name1	The XML XPath name for retrieving the value from the payload
XPath Expression1	The XML XPath expression for retrieving the value from the payload.
XPath Name2	The XML XPath name for retrieving the value from the payload.
XPath Expression2	The XML XPath expression for retrieving the value from the payload.
XPath Name3	The XML XPath name for retrieving the value from the payload.
XPath Expression3	The XML XPath expression for retrieving the value from the payload.
Correlation Tab	-

Table 3-2 (Cont.) Document Definition Parameters for a Custom Document

Parameter	Description
Correlation From XPath Name	The name of the correlation property for initiating the correlation.
Correlation From XPath Expression	The XML XPath for retrieving the value from the payload to initiate the correlation.
Correlation To XPath Name	The name of the correlation property for the correlation.
Correlation To XPath Expression	The XML XPath for retrieving the value from the payload for the correlation.
Flat Tab	-
Identification Start Position	Used in combination with the end position to retrieve a value from the payload between the start and end positions
Identification End Position	Used in combination with the start position to retrieve a value from the payload between the start and end positions
Identification Value	A value between the start and end positions
Apps Tab	-
Document	The name of the internal application document.
Action	A sub-classification within the document.
XSLTFile	The name of the XSLT file.

3.2.3.1 How to Configure the XPath Expression for a Custom XML Document

The XPath expression identifies a Custom XML document. You configure the XPath expression when you specify the document type parameters.

The options when configuring an XPath expression are as follows:

- Option 1: Specify the XPath and the Matching Value

- Option 2: Check for the Existence of a Node

- Option 3: Check the Value of an Attribute

3.2.3.1.1 Option 1: Specify the XPath and the Matching Value

Assume that the transaction ID is 12345. Set the parameters as follows:

Field	Value
Identification Value	12345
Identification Expression	//*[local-name() = 'TransactionID']/text()

Oracle SOA Suite for healthcare integration compares the value of **Identification Expression** in the payload to the value specified in **Identification Value**. If the values match, then the document is identified successfully and the corresponding document type and document protocol version are used to identify the endpoint. **Example -**

Specify the XPath and the Matching Value shows an excerpt of the XML payload for this option.

Example - Specify the XPath and the Matching Value

```
<?xml version="1.0" encoding="UTF-8" ?>
<Message xmlns:ns1="http://www.example1.org" xmlns:ns2="http://www.example2.org"
  xmlns="http://www.example3.org"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:ns="http://www.example4.org">
  <MessageHeader>
    <Source>201944019</Source>
    <Destination>205704856</Destination>
    <TransactionID>123456</TransactionID>
    <Version>1-0-0</Version>
  </MessageHeader>
  <Body>
    <ns:Case xsi:schemaLocation="http://www.example4.org" ns1:caseCategoryID="1">
        <ns1:OfficialProvisionNumber>String</ns1:OfficialProvisionNumber>
    </ns:Case>
  </Body>
</Message>
```

3.2.3.1.2 Option 2: Check for the Existence of a Node

Assume that you are checking for the existence of a node called `registerCommand`. Set the parameters as follows:

Field	Value
Identification Value	*Leave blank.*
Identification Expression	/*[local-name()='envelope']/body/transaction/command/ *[local-name()='registerCommand']

When the **Identification Value** field is left blank, Oracle SOA Suite for healthcare integration checks for the node identified in **Identification Expression**. If a node in the payload matches, then the document is identified successfully. Example - Check for the Existence of a Node shows an excerpt of the XML payload for this option.

Check for the Existence of a Node

```
<uccnet:envelope xmins:eanucc="http://www.ean-ucc.org/schemas/1.3/eanucc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:uccnet="http://www.uccnet.org/schemas/2.2/uccnet"
    communicationVersion="2.2"
  xsi:schemaLocation="http://www.uccnet.org/schemas/2.2/uccnet
  http://www.testregistry.net/xmlschema/uccnet/2.2/Envelope.xsd">
  <messageHeader>
    <messageIdentifier>
      <value>791:1_EB3CDC749A1F2BABE03014906CC4605A</value>
    </messageIdentifier>
    <userId>oraclesupXSD</userId>
    <representingParty>
      <gin>0060974050142</gin>
    </representingParty>
  </messageHeader>
  <body>
    <transaction>
      <entityIdentification>
        <uniqueCreatorIdentification>856</uniqueCreatorIdentification>
```

```
                    <globalLocationNumber>
                      <gin>0060974050142</gin>
                    </globalLocationNumber>
               </entityIdentification>
               <command>
                    <uccnet:registerCommand>
                      <registerCommandHeader type="ADD" />
                    </uccnet:registerCommand>
               </command>
          </transaction>
     </body>
</uccnet:envelope>
```

3.2.3.1.3 Option 3: Check the Value of an Attribute

Assume that the value of the country attribute is **US**. Set the parameters as follows:

Field	Value
Identification Value	US
Identification Expression	//*/@country

Oracle SOA Suite for healthcare integration compares the value of the country attribute to the value set for **Identification Value**. If the values match, then the document is identified successfully. **Example - Check the Value of an Attribute** shows an excerpt of the XML payload for this option.

Example - Check the Value of an Attribute

```
<?xml version="1.0" encoding="windows-1252" ?>
<MyAddress country="US" xmlns="http://www.example.org"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="PO.xsd">
  <name>B2B Buyer</name>
  <street>100 Oracle Parkway</street>
  <city>Redwood City</city>
  <state>CA</state>
  <zip>94065</zip>
</MyAddress>
```

3.3 Using the HL7 Document Protocol

Oracle SOA Suite for healthcare integration implements the Health Level 7 (HL7) version 2.x to exchange documents containing health care information using the Generic exchange or MLLP exchange.

When using HL7, the standard Oracle SOA Suite for healthcare integration features, such as validation, translation, automatic generation of outbound envelope headers, and acknowledgments, are available.

For information about the organization that created and maintains the HL7 standards, go to http://www.hl7.org.

3.3.1 What You Might Need to Know About HL7 Document Version Parameters

When you create an HL7 document version, you can set various parameters. Figure 3-7 shows document version parameters for an HL7 document.

Figure 3-7   Document Version Parameters for an HL7 Document
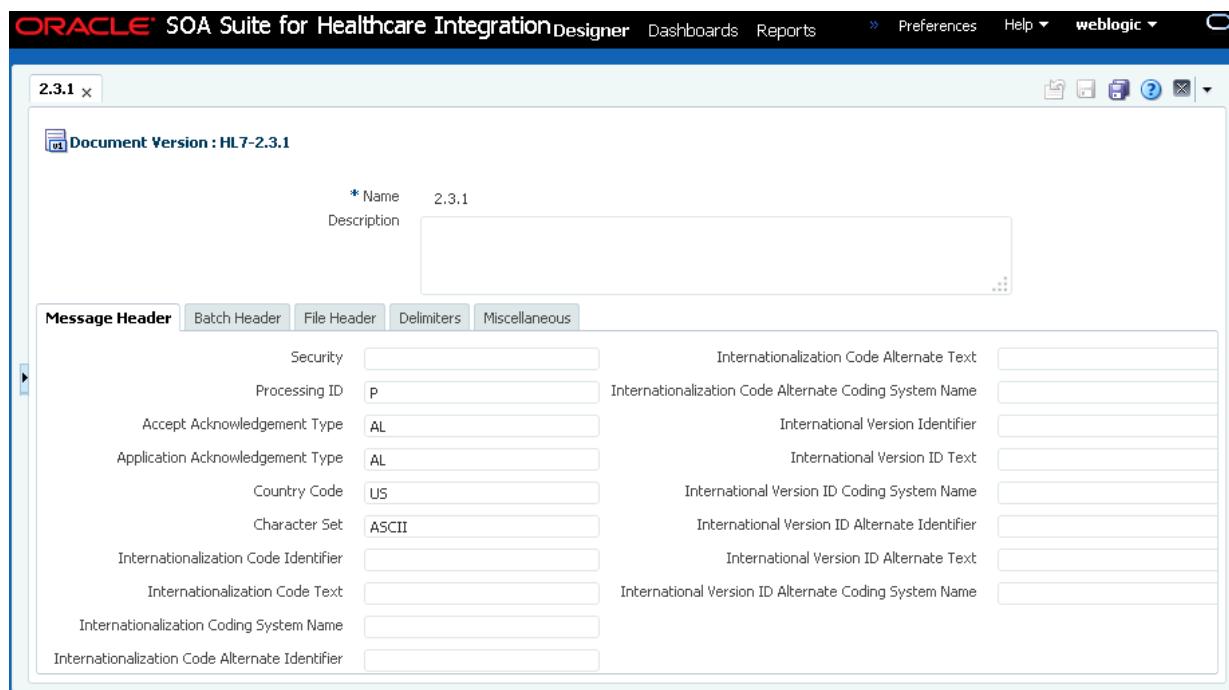
Table 3-3 describes the document version parameters for an HL7 document.

Table 3-3   Document Version Parameters for an HL7 Document

Parameter	Description
Message Header Tab	-
Security	In some applications of HL7, this field is used to implement security features.
Processing ID	MSH.11 - This field is used to decide whether to process the message as defined in HL7 Application (level 7) processing rules. The first component defines whether the message is part of a production, training, or debugging system (refer to HL7 table 0103 - Processing ID for values). The second component defines whether the message is part of an archival process or an initial load (refer to HL7 table 0207 - Processing mode for values). This allows different priorities to be given to different processing modes.
Accept Acknowledgement Type	Sets the conditions under which application acknowledgments are required to be returned in response to the message. The value **AL** (always) is supplied. Oracle SOA Suite for healthcare integration checks the payload (MSH.15) of an incoming message to see if an ACK has to be generated. In some HL7 Systems, MSH.15 is not sent in the payload at all and it is expected that an ACK is still sent.
Application Acknowledgment Type	MSH.16. The value **AL** (always) is supplied.
Country Code	Sets the country of origin for the message. The value US is supplied.
Character Set	Sets the character set for the entire message. The value ASCII is supplied.

Table 3-3   (Cont.) Document Version Parameters for an HL7 Document

Parameter	Description
Internationalization Code Identifier	MSH.19
Internationalization Code Text	MSH.19
Internationalization Coding System Name	MSH.19
Internationalization Code Alternate Identifier	MSH.19
Internationalization Code Alternate Text	MSH.19
Internationalization Code Alternate Coding System Name	MSH.19
International Version Identifier	MSH.12
International Version ID Text	MSH.12
International Version ID Coding System Name	MSH.12
International Version ID Alternate Identifier	MSH.12
International Version ID Alternate Text	MSH.12
International Version ID Alternate Coding System Name	MSH.12
Batch Header Tab	-
Create Batch Header	Select the box to create batch headers.
Batch Header Ecs File	Click the **Browse** button to find an ecs file to override the standard file. If not provided, the provided default file is used.
Batch Security	BHS.8
Batch Date	BHS.7. The system date-time stamp is supplied (#SystemDateTime(CCYYMMDDHHMM)#).
File Header Tab	-
Create File Header	Select the check box to create file headers.
File Header Ecs File	Click the **Browse** button to find an ecs file to override the standard file. If not provided, the provided default file is used.
File Security	FHS.8
File Date	FHS.7. The system date-time stamp is supplied (#SystemDateTime(CCYYMMDDHHMM)#).

Table 3-3   (Cont.) Document Version Parameters for an HL7 Document

Parameter	Description
Delimiters Tab	Click **Select Hexadecimal Characters** next to any of the delimiter fields to provide values.
Element Delimiter	A single character that follows the segment identifier and separates each data element in a segment except the last. The value **0x7c** is supplied.
Escape Character	The value **0x5c** is supplied.
Repeating Separator	A service character used to separate adjacent occurrences of a repeating data element, or to separate multiple occurrences of a field.The value **0x7e** is supplied.
Segment Delimiter	A syntax character indicating the end of a segment (a logical grouping of data fields) within a message. The value **0x0d** is supplied.
Subcomponent Delimiter	The value 0x26 is supplied.
Subelement Delimiter	The value 0x5e is supplied.
Miscellaneous Tab	-
Ignore Envelope Parameters	Use this option to provide a list of envelope elements, separated by commas, to be ignored during look-up validation. For an HL7 agreement, the possible values include MessageSendingApp, MessageReceivingApp, MessageSendingFacility, and MessageReceivingFacility.
Ack Mode	Select this option to specify whether to send a single acknowledgment or multiple acknowledgments for inbound batched HL7 messages.

3.3.2 What You Might Need to Know About HL7 Document Type Parameters

When you create an HL7 document type, you can set various parameters. Figure 3-8 shows the document type parameters for an HL7 document.

Figure 3-8   Document Type Parameters for an HL7 Document
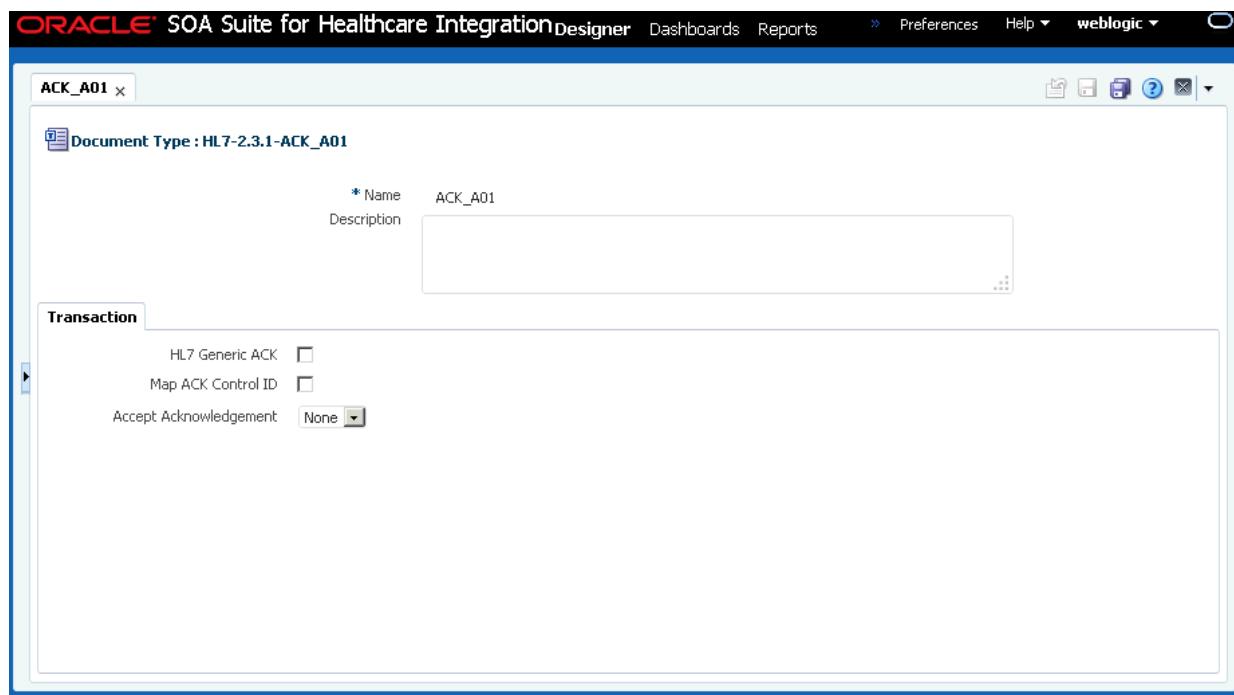
Table 3-4 describes the document type parameters for an HL7 document.

Table 3-4   Document Type Parameters for an HL7 Document

Parameter	Description
Transaction Tab	-
HL7 Generic ACK	If selected, Oracle SOA Suite for healthcare integration sends a generic ACK. **Note:** The HL7 standard does not specify any Ack generation rules for the errors on file or batch header - only for the message. Oracle B2B does not generate acks for file and batch header errors for HL7 inbound messages (only error reports are generated).
Map ACK Control ID	Select to enable mapping the MSH.10 of the business message to the MSH.10 of the acknowledgment. **Note:** This Map ACK Control ID parameter is for the functional ACK.
Accept Acknowledgement	A functional acknowledgment is generated when MSH.15 has no value. Select **None** to take no action. Acknowledgment generation is dependent on the value in MSH.15 of the business message. Select **AL** (always) to generate the acknowledgment under any conditions. Select **ER** (error/reject) to generate the acknowledgment when the message errors or is rejected. Select **SU** (successful completion) to generate the acknowledgment when the message is successfully processed.

For information about how to override document parameters at individual endpoint level see Overriding Document Parameters at the Endpoint Level.

3.3.3 What You Might Need to Know About HL7 Document Definition Parameters

When you create an HL7 document definition (see Creating Document Definitions for more information on creating document definitions), you can set various parameters. Figure 3-9 shows document definition parameters for an HL7 document.

Figure 3-9   Document Definition Parameters for an HL7 Document
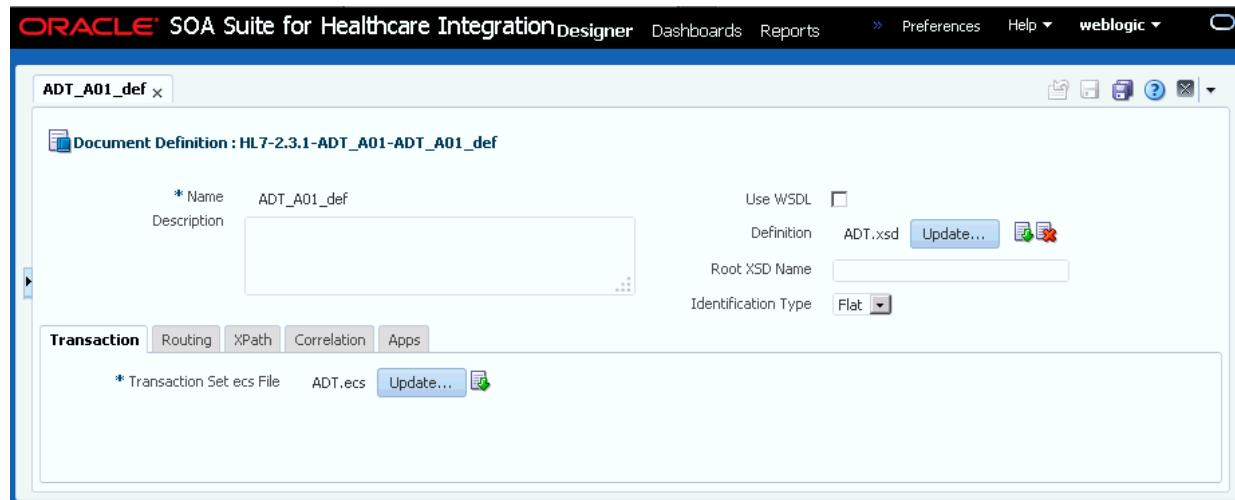
Table 3-5 describes the document definition parameters for an HL7 document.

Table 3-5   Document Definition Parameters for an HL7 Document

Parameter	Description
Transaction Tab	-
*Transaction Set ecs File	Click the **Browse** button to find the ecs file.
Routing Tab	-
Document Routing ID	Sets the consumer name to the back-end application
XPath Tab	See How to Configure the XPath Expression for a Custom XML Document, for more information.
XPath Name1	The XML XPath name for retrieving the value from the payload
XPath Expression1	The XML XPath expression for retrieving the value from the payload
XPath Name2	The XML XPath name for retrieving the value from the payload
XPath Expression2	The XML XPath expression for retrieving the value from the payload
XPath Name3	The XML XPath name for retrieving the value from the payload
XPath Expression3	The XML XPath expression for retrieving the value from the payload
Correlation Tab	-

Table 3-5   (Cont.) Document Definition Parameters for an HL7 Document

Parameter	Description
Correlation From XPath Name	The name of the correlation property for initiating the correlation.
Correlation From XPath Expression	The XML XPath for retrieving the value from the payload to initiate the correlation.
Correlation To XPath Name	The name of the correlation property for the correlation.
Correlation To XPath Expression	The XML XPath for retrieving the value from the payload for the correlation.
Apps Tab	-
Document	The name of the internal application document.
Action	A sub-classification within the document.
XSLTFile	The name of the XSLT file.

3.3.4 What You Might Need to Know About Using HL7

No business message is produced for an HL7 immediate acknowledgment (transport-level acknowledgment).

Negative acknowledgment messages indicating errors in an HL7 exchange might be truncated because of the 80-character length limitation in HL7 versions 2.1 through 2.5.

3.4 Creating Document Definitions

A document definition specifies the document protocol—the document protocol version and document type—that is used to validate the message. The document definition can be an ECS file, in the case of HL7 messages, or an XSD/DTD, in the case of XML messages.

The same document definition is used by participating endpoints in a transaction. It must adhere to the standards for document protocols, protocol versions, and document types.

> **Note:**
>
> To ensure that the document definition conforms to standards, you can use Oracle Document Editor to create the document guideline files and then use the Oracle SOA Suite for healthcare integration user interface to import those files when creating the document definition.

After creating transaction set files by using Oracle Document Editor, you use the Oracle SOA Suite for healthcare integration user interface to create the document definition and import the transaction set files.

Note:

The document version, document type, and document definition are not editable after they are created. You must delete the specific document element (version, type, or definition) and create a new one. Updating the document elements after creation can lead to metadata inconsistency, metadata validation issues, and runtime errors.

To create a document definition:

1. Log on to Oracle SOA Suite for healthcare integration. The Oracle SOA Suite for healthcare integration user interface opens with the Configuration tab selected.

2. In the left-hand navigation panel, expand **Document Protocol**. This displays the list of supported protocols, which are Custom and HL 7 V2.x.

3. Select one of the document protocols, for example HL7, as shown in Figure 3-10, and click the **Create** button. This displays the Create Document Version window.

Note:

You can also right-click the document protocol name and select **Create** from the shortcut menu.

Figure 3-10   Selecting a Document Protocol
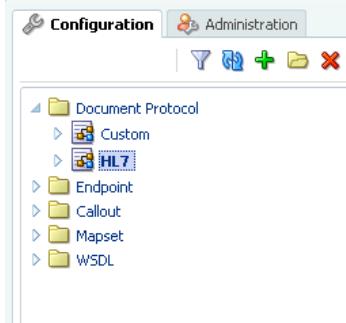
Note:

In the Create Document Protocol Version dialog box, enter a protocol version name, for example, 2.3.1.

4. Enter the document protocol version name in the **Name** field.

5. Specify the document version parameters as applicable, and click **OK**.

   The version is used for document identification and can be case sensitive. Use a fixed syntax that conforms to the protocol standards.

   Figure 3-11 shows the document protocol version page for an HL7 V2.3.1 document.

Figure 3-11   Entering Document Protocol Version Information
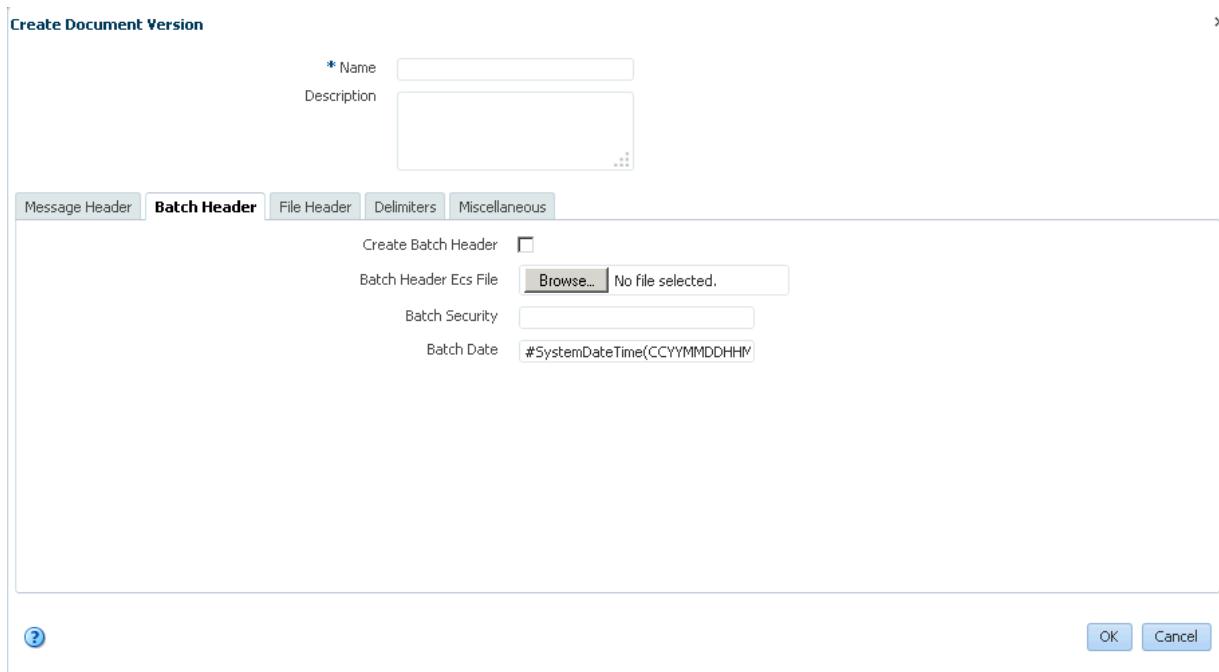
For parameter descriptions, see Table 3-3.

6. Click the newly created Version name, and then click the **Create** button to display the Create Document Type window. Alternatively, you can right-click the version name and select **Create** from the shortcut menu.

7. Enter a document type name, specify document type parameters as applicable, and then click **OK**.

   Figure 3-12 shows the document type parameters page for an HL7 V2.3.1 document.

Figure 3-12   Entering Document Type Parameter Information
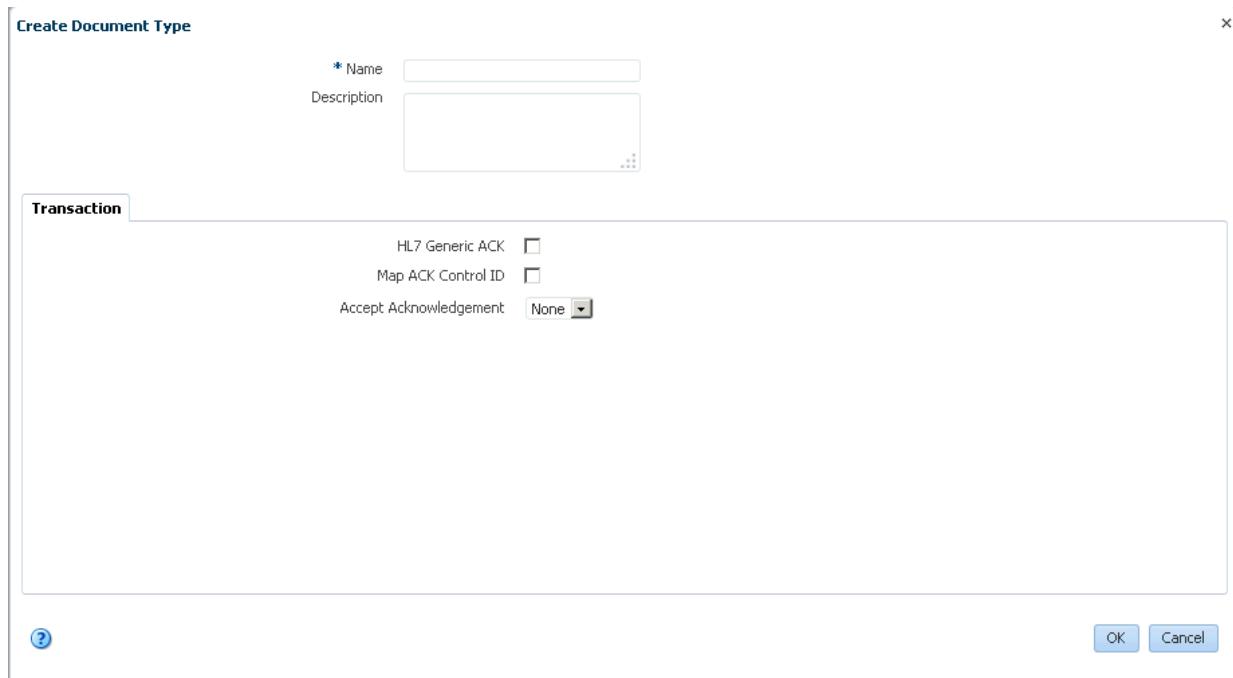
For document type parameter descriptions, see Table 3-4.

8. With the new document type name selected, click **Create** to display the Create Document Definition window.

9. Enter a document definition name and do the following:

a. Browse for an optional definition (XSD) file for any of the document protocols.

b. Browse for the required transaction set ECS file for HL7 or positional flat file.

c. Specify document definition parameters as applicable and click **OK**.

Figure 3-13 shows the document definition parameters page for an HL7 V2.3.1 document.

Figure 3-13   Entering Document Definition Parameter Information
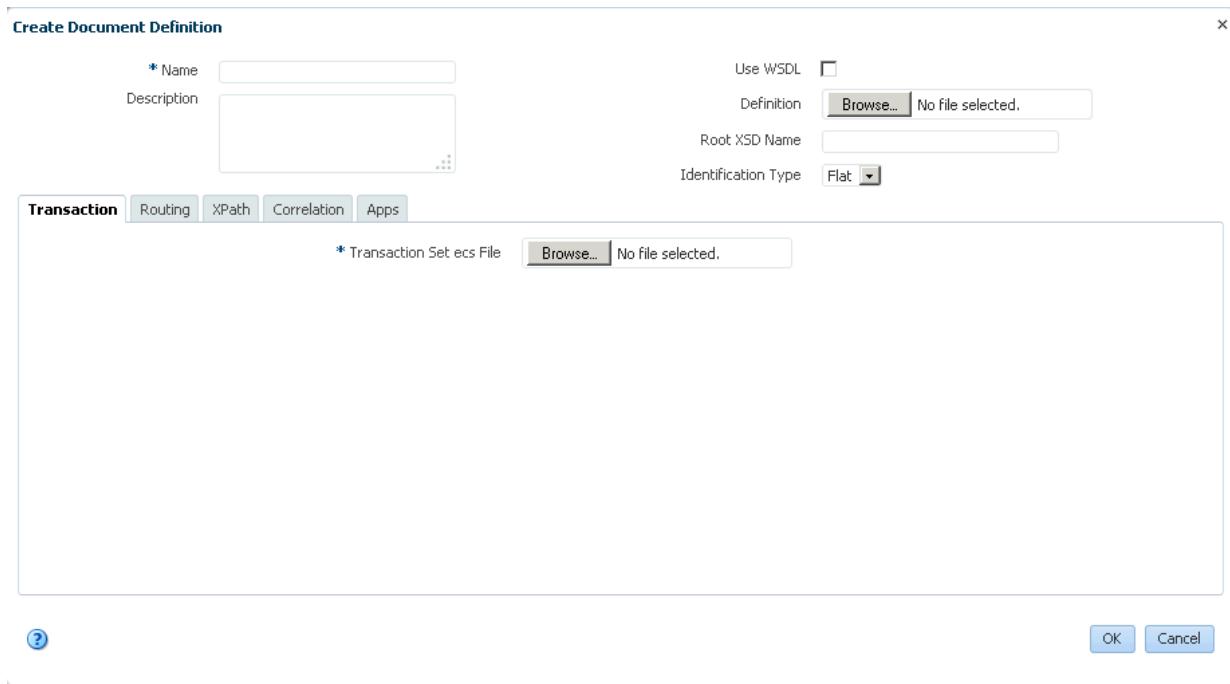
For definition parameter descriptions, see the following:

- Table 3-2

- Table 3-5

Note:

You can also use the post install script, install the standard library, and import hl7 doctypes file to create a document tree in Healthcare console.

3.5 Deleting a Document Definition

To delete a document definition, select the document definition name and click **Delete**. Alternatively, you can select the definition name, right-click, and select **Delete** from the shortcut menu.

4

Working with Endpoints

This chapter describes the concept of an endpoint in Oracle SOA Suite for healthcare integration, and provides instructions for creating and configuring endpoints for healthcare integration applications.

This chapter contains the following topics:

- Introduction to Endpoints

- Creating Endpoints

- Associating an Endpoint with a Document

- Enabling Sequencing for an MLLP Endpoint

- Enabling SSL/TLS Support for MLLP Endpoints

- Handling Actionable Errors for an MLLP Endpoint

- Deleting an Endpoint

- Working with the Endpoint Window

- Healthcare and Oracle Managed File Transfer Integration

4.1 Introduction to Endpoints

In Oracle SOA Suite for healthcare integration, endpoints are communication channels from where predefined documents are sent or received. Endpoints define how documents are exchanged with an external system, specifying the location, transport protocol, documents to be exchanged, and other configuration parameters.

An endpoint can be a URL, folders, or path, among others. Based on the direction of the message, an endpoint can be inbound, outbound, or both. For example, when Oracle SOA Suite for healthcare integration reads from a directory, the directory is the inbound endpoint. Conversely, when Oracle SOA Suite for healthcare integration writes to or sends messages to a directory, the directory is the outbound endpoint. Also, an MLLP endpoint can be used both for receiving and sending messages.

For Oracle SOA Suite for healthcare integration, you must associate an endpoint with document definitions and enable the endpoint to be able to start sending and receiving messages.

Figure 4-1 displays a sample endpoint, which is yet to be associated with a document definition.

Figure 4-1   Oracle SOA Suite for healthcare integration Sample Endpoint
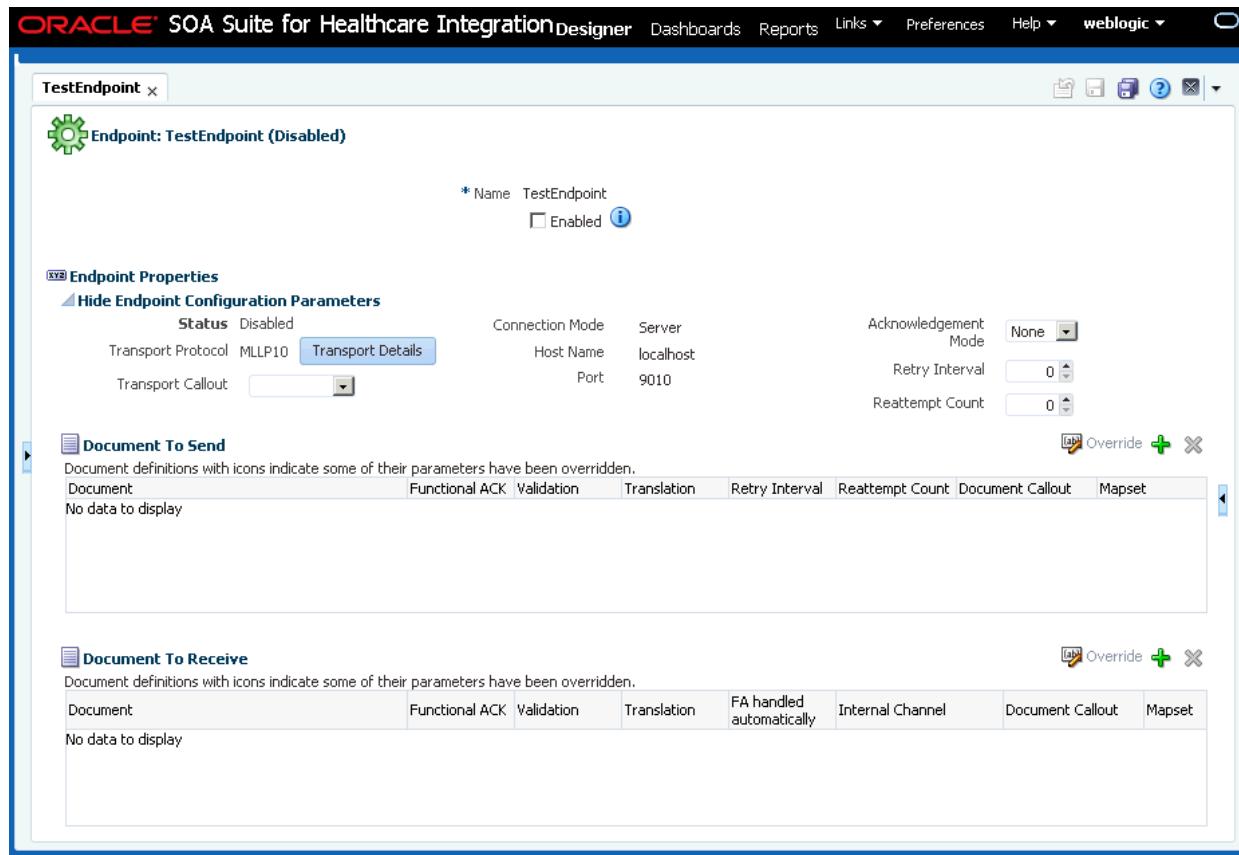
4.2 Creating Endpoints

The Oracle SOA Suite for healthcare integration user interface provides an Endpoints page where you can create and configure endpoints. This procedure describes how to create endpoints using the interface.

You can create endpoints both for bidirectional (MLLP) and single-directional (FILE, JMS) transport protocols. Bidirectional protocols enables you to send or receive response messages or functional acknowledgements (FAs) by using the same endpoint. However, single-directional protocols must be configured to send and receive documents; only then you can send or receive messages or FAs per endpoint.

To create endpoints with bidirectional transport protocol (MLLP):

1.   Log on to the Oracle SOA Suite for healthcare integration user interface.

2.   In the **Configuration** tab under the **Design** tab, click the **Endpoint** folder and then click the **Create** button as shown in Figure 4-2.

Figure 4-2 Create Endpoint Button
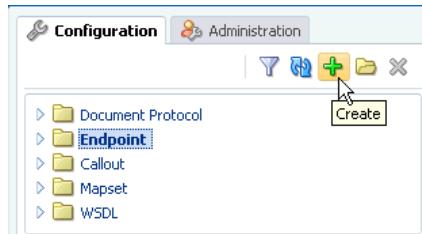
3. In the **Create** window, enter the following and click **OK**, as shown in Figure 4-3:

    *   **Name**: Name of the endpoint.

    *   **Transport Protocol**: Transport protocol for the sending or receiving messages. In this case, select **MLLP10**.

    *   **Connection Mode**: Server or Client. If the endpoint is configured as server, Oracle SOA Suite for healthcare integration engine starts listening on a port and waits for a client to connect to it. In general, the server connection mode is for inbound case. When configured as client, the engine connects to hostname and port of a remote computer or device. In general, this is for an outbound case.

        > **Note:**
        >
        > Oracle Healthcare supports multiple client connections for a single server channel. So, multiple clients can connect to the same server and exchange data and receive FAs.
        >
        > If the message is passed to the back-end application, then the message can be sent in the following ways:
        >
        > –   Oracle Healthcare passes the endpoint details to the back-end application using dynamic IP address, and the application populates the same values back into outbound dynamic IP header. Oracle Healthcare makes use of this dynamic IP header to pick the correct connection.
        >
        > –   Clients can set the `replytoMsgId` property in the properties of the acknowledgement or response message and then connection details can be fetched from the request message.

    *   **Host Name**: In case of an MLLP Server endpoint, it should be name or IP address of the computer hosting Oracle SOA Suite, and in the case of an MLLP Client endpoint, it should be the remote host name or device name. Typically, this should be `localhost`. However, the when localhost is specified, the WLS server listen address is used, which may not be what is expected.

        > **Note:**
        >
        > When using NIO mode, you must specify the IP address or the computer name instead of `localhost`.

    *   **Port**: Port number should be more than 500. If the connection mode is set to Server, then the port must be a valid TCP port number. If the connection mode

is set to Client, then the port must be the same as the port used on the MLLP server.

This creates the endpoint and the endpoint is displayed in the right panel of the Oracle SOA Suite for healthcare integration user interface.

Figure 4-3 Specifying Endpoint Parameters
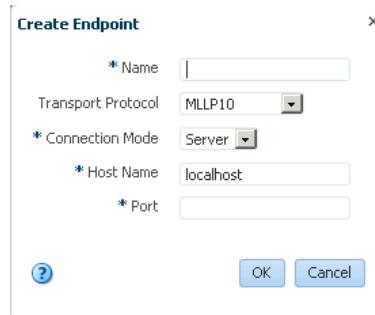
To create endpoints with single-directional transport protocol (FILE):

1. Log on to the Oracle SOA Suite for healthcare integration user interface.

2. In the **Configuration** tab under the **Design** tab, click the **Endpoint** folder and then click the **Create** button.

3. In the **Create** window, enter the following and click **OK**, as shown in Figure 4-4:

   - **Name**: Name of the endpoint.

   - **Transport Protocol**: Transport protocol for the sending or receiving messages. In this case, select **FILE**.

   - **Direction**: Inbound or Outbound based on your requirement. If the endpoint is configured as inbound, then it can receive response messages or FAs from other endpoints. Conversely, if the endpoint is configured as outbound, it can send messages or FAs.

   - **Folder Name**: An absolute directory path is recommended and this folder does not contain the inbound or outbound messages or FAs. Inbound messages are expected in this folder, and outbound messages or FAs must be delivered here.

This creates the endpoint and the endpoint is displayed in the right panel of the Oracle SOA Suite for healthcare integration user interface.

Figure 4-4 Specifying Endpoint Parameters
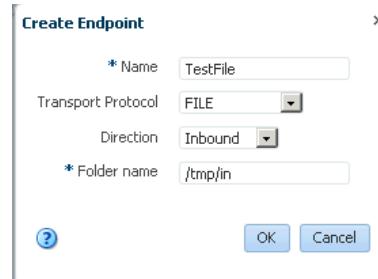
Note:

After a single-directional endpoint (inbound/outbound) is created, then it can be edited later to add inbound or outbound configuration by clicking the **Configure** link.

Note:

See Creating Endpoints with Different Transport Protocols for more information on creating endpoints with different transport protocols.

4.2.1 Configuring Channels in an Endpoint with Single-Directional Protocols

When a bidirectional transport protocol (such as MLLP) is used to create an endpoint, the endpoint can handle both request and reply (inbound and outbound) or ACK. However, in the case of a single-directional transport protocol (such as File or FTP), typically, you can only configure either the inbound or the outbound channel, but not both together.

Oracle SOA Suite for healthcare integration, in the case of single-directional protocols, allows you to configure two communication channels (one for sending and one for receiving messages) as a single entity (endpoint) for management and monitoring. This concept allows you to enable or disable, view the supported documents, and display related monitoring data (message counts and actual messages) for the related channels at the same time.

Using this feature, when you use a single-directional transport protocol, you must configure only one channel. The configuration for the other channel (for the reverse direction) is optional. The Oracle SOA Suite for healthcare integration console, in this case, displays that one channel has been configured to send or receive messages, but the other channel has not been configured yet as shown in Figure 4-5.

Figure 4-5   Undefined Secondary Channel
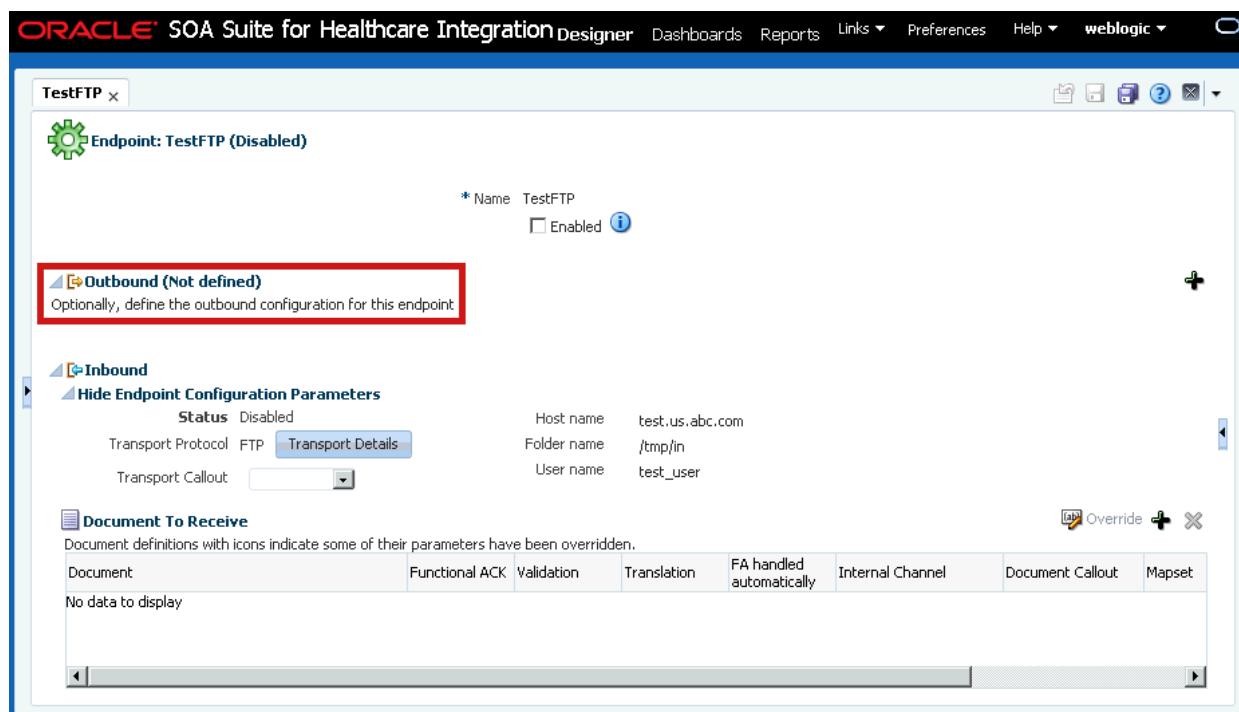
You can configure the other channel by clicking the **+** button in the undefined channel section, as shown in Figure 4-6, and providing the required channel details in the configuration dialog box.

Figure 4-6   Defining the Secondary Channel
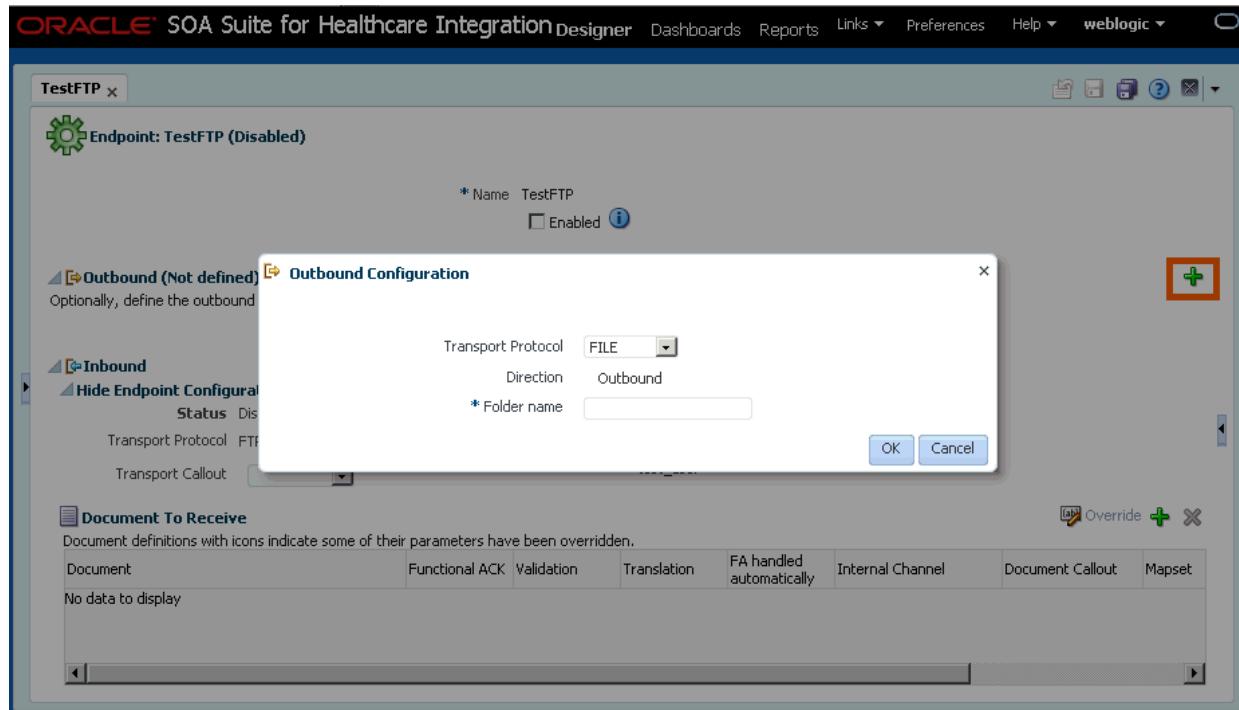
The transport protocol and other settings are defined per channel. You can use different single-directional transport protocols for the two channels. For example, the

one channel can use FTP, whereas the other channel can use File. The only requirement is that both the protocols used should be single directional.

After you have defined both the communication channels and associated documents with the channels (see Associating an Endpoint with a Document), the endpoint appears as Figure 4-7.

Figure 4-7   Single-Directional Endpoint with Both Channels Configured
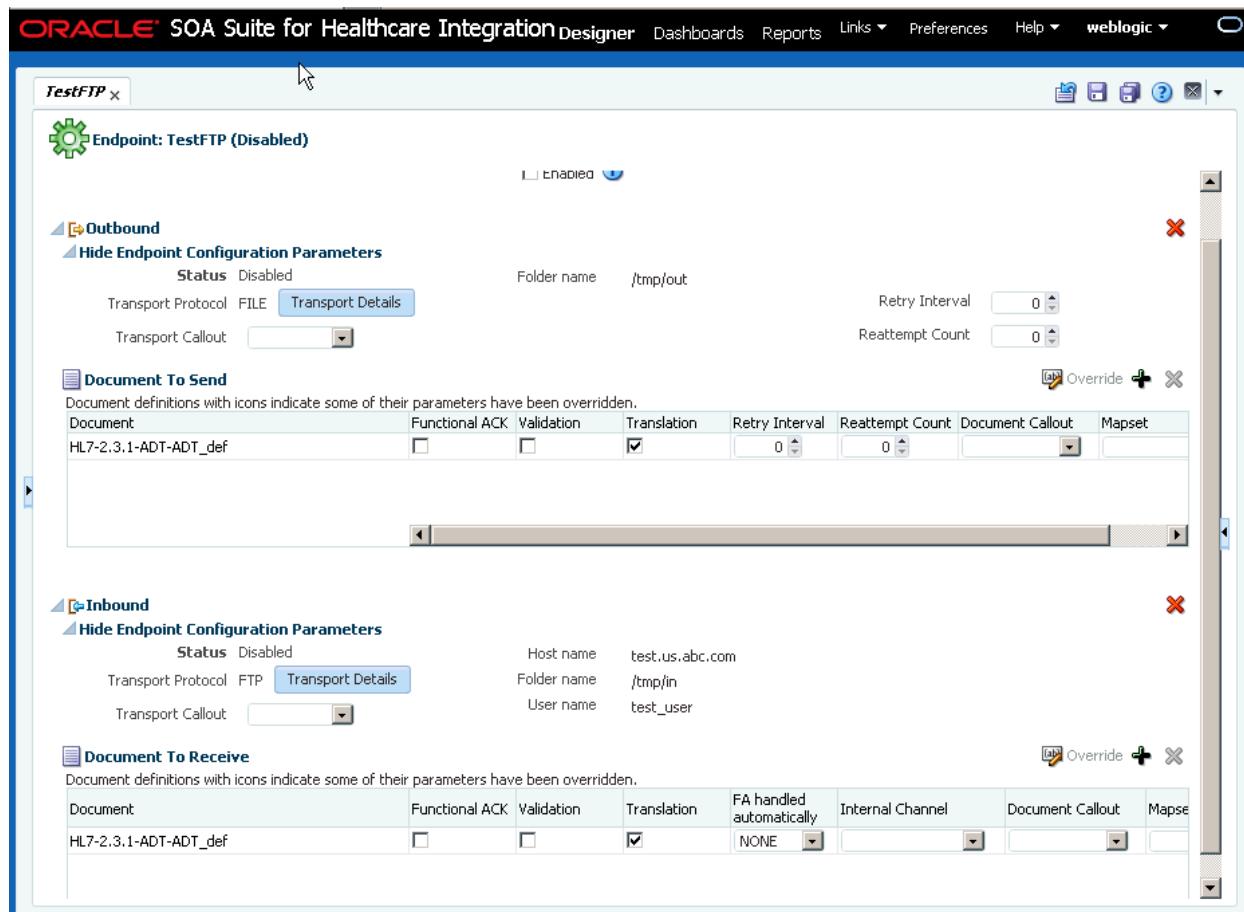
You can edit the configuration of either of the channels if required. You can also delete the configuration of either of the channels by using the Delete links in the relevant channel section.

4.3 Associating an Endpoint with a Document

After you have created an endpoint, you must associate it with a document to enable the endpoint to send or receive messages. You can configure an endpoint to send or receive messages or both.

To associate a bidirectional endpoint with a document:

1. Open the required endpoint.

2. Configure the endpoint to send or receive messages:

    a. In the Send or Receive section of the endpoint, click the **+** button to display the Document selection window.

b. Select the required document definition from available document hierarchy, for example, ADT_A04_def. The document definition gets associated with the endpoint. You can specify whether you require functional acknowledgment, validation, or translation of the endpoint as well as required internal delivery channel, transport or agreement level callouts, and mapsets.

Note:

you can also drag-and-drop the document definition to the Send or Receive section.

Figure 4-8 displays an endpoint associated with a document definition.

Figure 4-8   Associating an Endpoint with Document Definitions

---
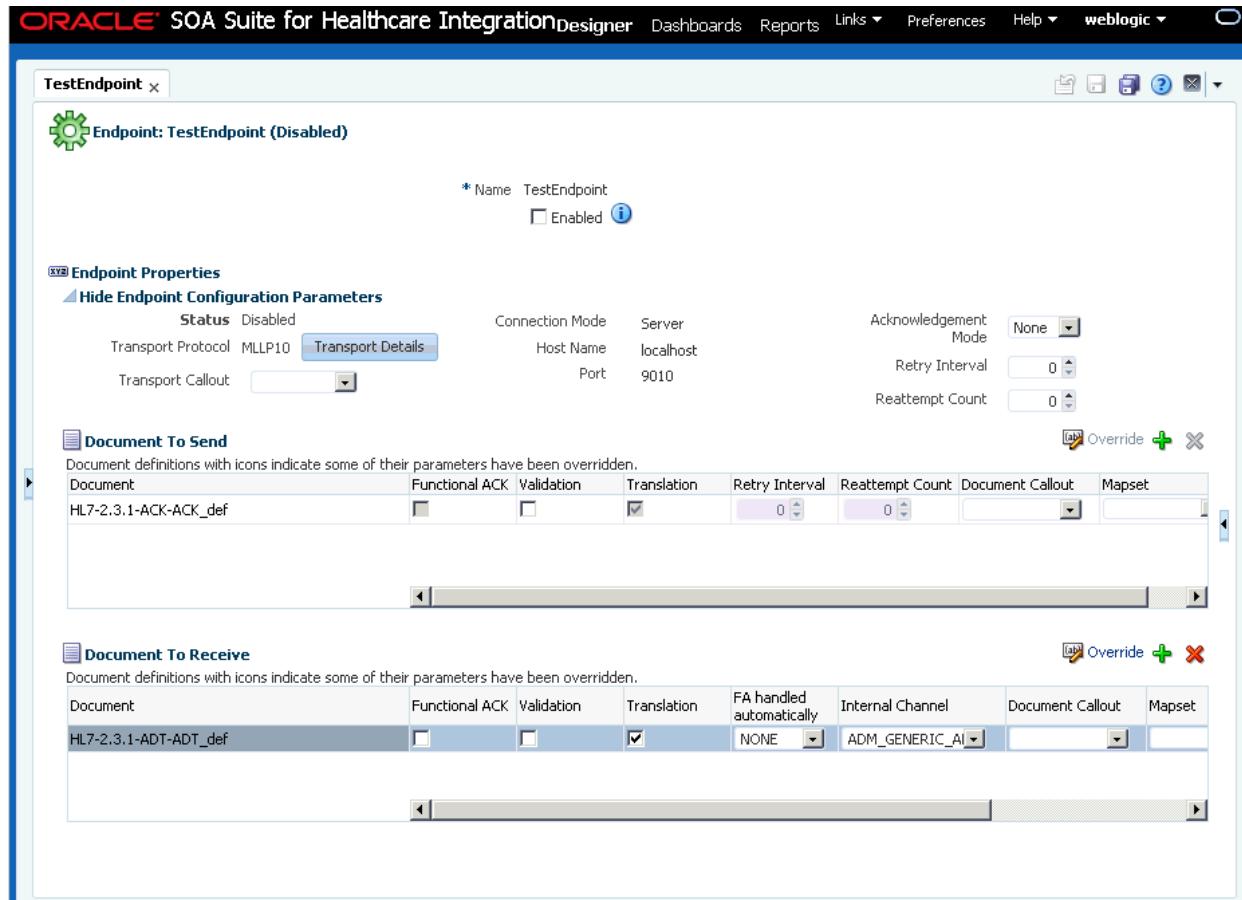
Note:

For FA, Functional ACK, Validation, Translation, Retry Interval, and Reattempt Count are disabled.

---

c. Select any of the following document configuration options:

Option	Description
Functional ACK	Select to enable the functional acknowledgment for success or error criterion
Validation	Select to enable validation of the document against the configured ECS file
Translation	Select to enable the translation of XML to native format and vice-versa

d. If any of the following have been defined for the endpoint, select them from the appropriate field: **Internal Channel**, **Document Callout**, **Mapset**, or **Composite**. For more information, see Chapter 6 Creating and Deploying Trading Partner Agreements in *Oracle Fusion Middleware User's Guide for Oracle B2B*.

> **Note:**
>
> Selecting the composite name and JMS internal delivery channel name are optional for an inbound document. Oracle Healthcare can support only one composite for a specific document definition. If multiple composites are available, the result is unpredictable because Oracle Healthcare sends messages to the one that is first registered to Oracle Healthcare during runtime.

3. Select the **Enabled** check box and click **Apply** to enable the endpoint for sending and receiving messages.The Apply operation sometimes could take about 30 to 60 seconds. This is due to the XSD/ECS creation and metadata validation.

> **Note:**
>
> After making any changes to an endpoint, you can right-click the endpoint name in the left-side panel and click **Refresh** to update the endpoint.

> **Note:**
>
> You can enable or disable an endpoint by selecting or deselecting the **Enabled** check box and clicking the **Apply** button.

Associating a single-directional endpoint with a document

To associate a single-directional endpoint, such as File, the procedure is similar to that of a bidirectional endpoint. However, there are a few differences:

- When creating the endpoint, based on the direction that you have selected, the Oracle SOA Suite for healthcare integration console opens the configuration option for that particular direction.

- After you follow the other steps listed in associating a bidirectional endpoint with a document, the endpoint gets attached to a document as shown in Figure 4-9.

Figure 4-9   Associating a Single-directional Endpoint with a Document
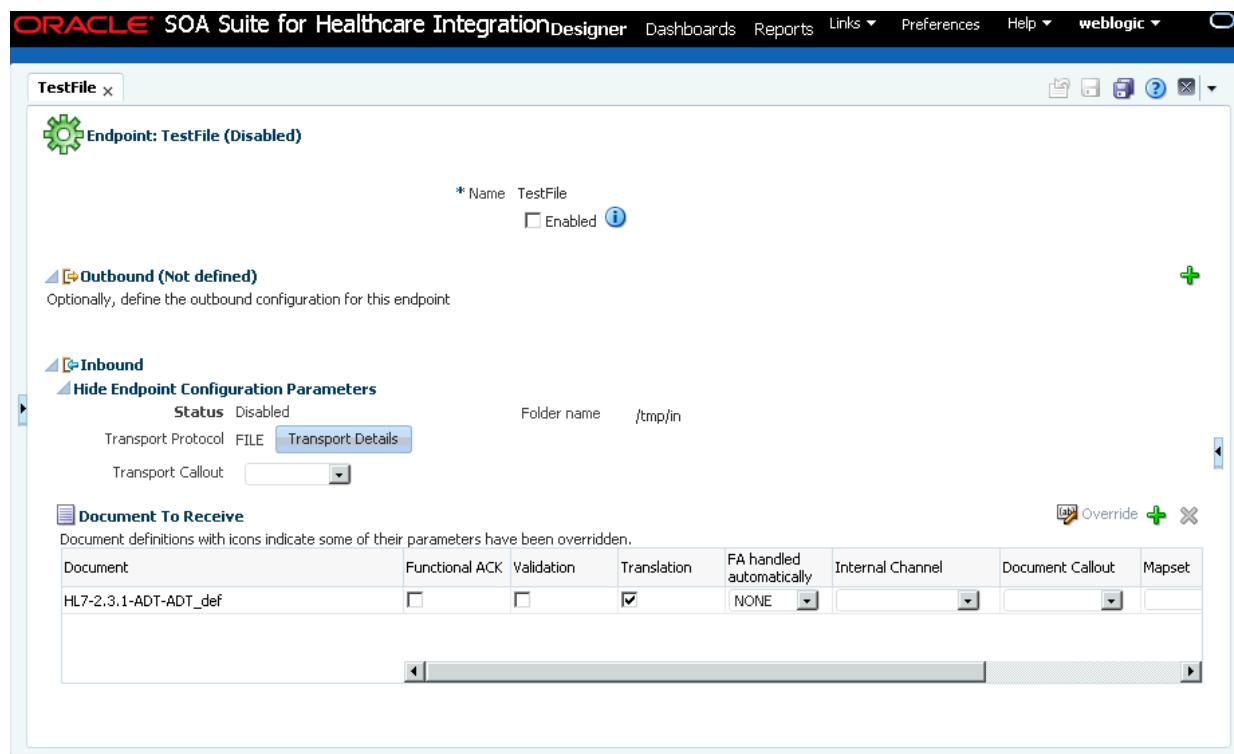
- If when creating the endpoint, you have specified the direction of the endpoint as inbound, you can define the outbound configuration for the endpoint on the same page by using the **Configure** link. This feature helps in tying two related single-directional endpoints (one for sending and one for receiving messages) as a single entity for better management and monitoring.

> **Note:**
>
> You can disassociate a document from an endpoint by selecting the specific document entry in the Documents section and then by clicking the **Delete** (X) button.

4.3.1 Overriding Document Parameters at the Endpoint Level

Oracle SOA Suite for healthcare integration enables you to override document parameters at individual endpoint level. Typically, documents are common for all endpoints. In that case, it is not possible to modify the Document Version and the Document Type parameters for a specific endpoint.

With the document parameter override feature, you can override the document parameters specific to an endpoint. For example, Oracle SOA Suite for healthcare integration sends a generic acknowledgement if the HL7 Generic ACK option is enabled in the document definition. So, all the endpoints using the same document definition send generic acknowledgements. However, if you want this feature to be enabled only for a selected set of endpoints, you must override the document definitions for those endpoints.

To override document parameters:

1. Open the required endpoint.

2. Click the document link that you want to override under the **Document To Send** or **Document To Receive** sections to display the document details dialog box. This dialog box has a tab each for the **Document Version** and the **Document Type**.

3. Click the **Document Version** tab to display all the sub-tabs related to Document Version, such as **Batch Header**, **Message Header**, **Delimiters**, **File Header**, and **Miscellaneous**, of the document that you have selected in the preceding step as shown in Figure 4-10.

Figure 4-10   Overriding Document Version Parameters
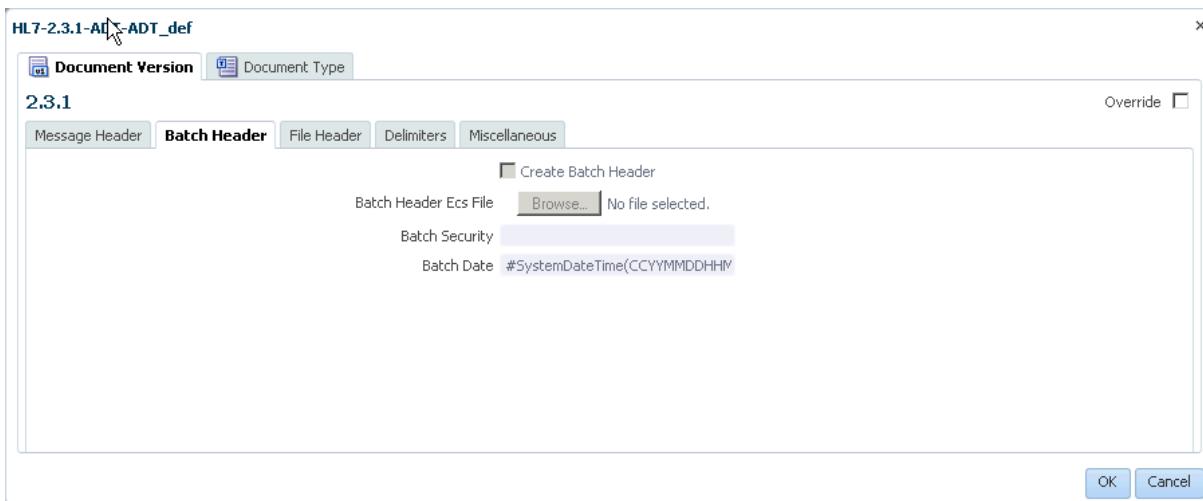
4. Select the **Override** check box to make the fields in each of these tab editable.

5. Make your changes. See What You Might Need to Know About HL7 Document Version Parameters to know more about the HL7 document parameters.

6. Click the **Document Type** tab to display the parameters related to the Document Type, such as **HL7 Generic ACK**, **Map ACK Control ID**, and **Accept Acknowledgement** as shown in Figure 4-11.

Figure 4-11   Overriding Document Type Parameters
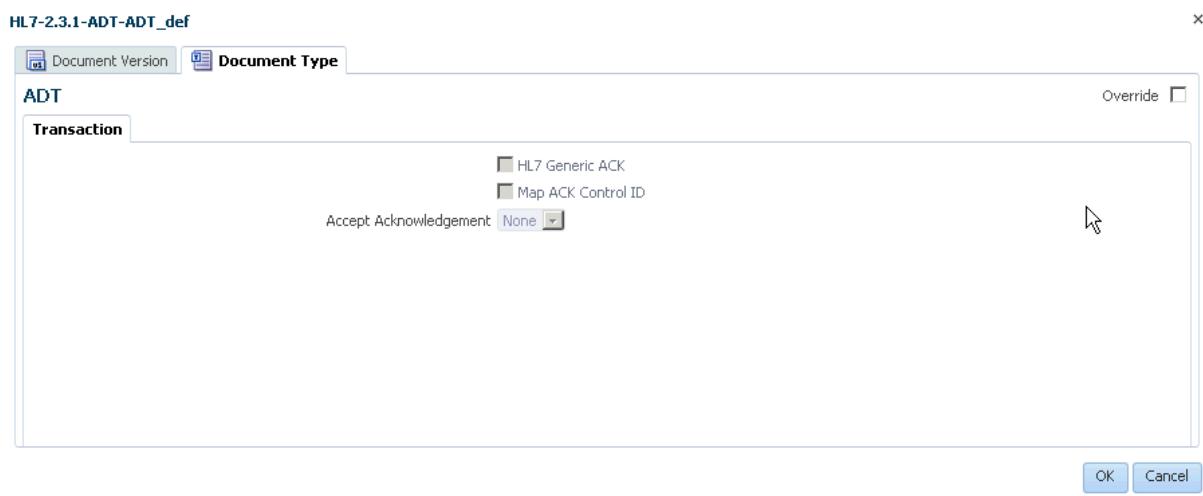
7. Select the **Override** check box to make the fields in each of these tab editable.

8. Make your changes. See What You Might Need to Know About HL7 Document Type Parameters to know more about the HL7 document parameters.

9. Click **OK**.

The overridden document definition is marked with a button next to it as shown in Figure 4-12.

Figure 4-12   Overridden Document Definition
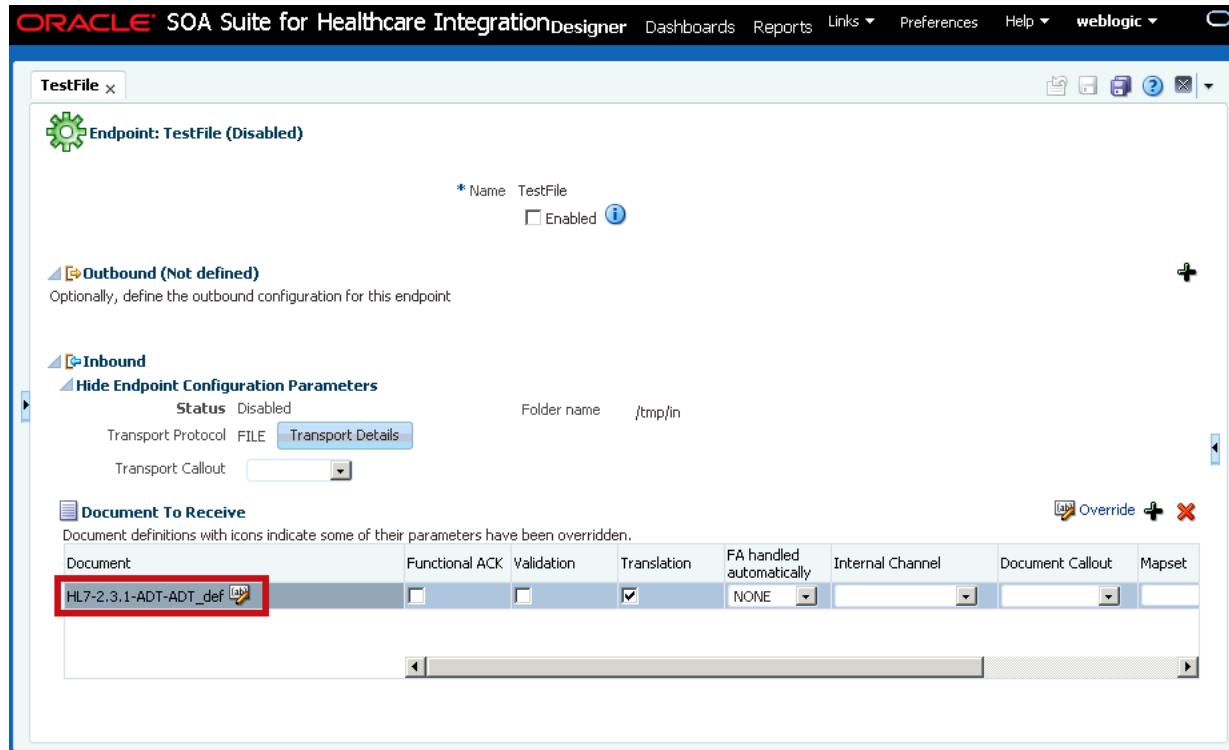
4.4 Enabling Sequencing for an MLLP Endpoint

After you associate a document with an endpoint, several options are available for configuring the endpoint.

- Acknowledgement (ACK) Mode: Select **Sync**, **Async**, or **None** for the mode in which the endpoint receives messages.

- Retry Interval: The interval between each attempt to retry message delivery.

- Reattempt Count: The number of times to retry message delivery.

- Transport Callout: The transport callout to be invoked after receiving or before sending any message. A callout can be selected only after the callout is created.

- Transport Protocol: Click the **Transport Details** button to customize the transport protocol parameters as shown in the following graphic.
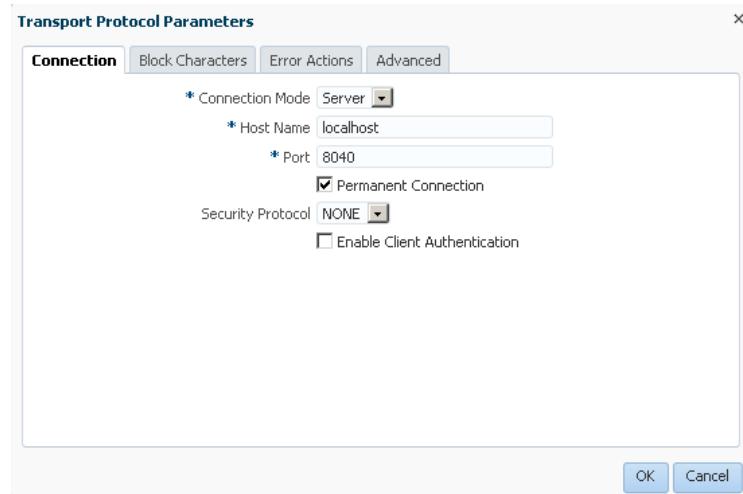
If HL7 messages over MLLP 1.0 are required to be sent in such a way that the next message is sent only after receiving a positive ACK for the current message, then the sequencing mode of MLLP 1.0 can be set to **OnetoOne** sequencing.

When associating an MLLP 1.0 endpoint with a document, you can select from the following sequencing modes for outbound messages from the **Advanced** tab of the **Transport Protocol Parameters** dialog box:

– **None**: Messages are dispatched in sequence without waiting for an ACK

– **OneToOne** (Default): Messages are sent in a sequenced manner, but the ACKs are not expected to carry the Control Numbers (the correlation is done without checking the control number in the ACK). In case of a negative ACK, the message sending is retried until either a positive ACK is received or the retry count is exhausted (at which point, the message goes into an error state.)

– **OneToOneMapping**: Messages are sent in a sequenced manner, but the ACKs *must* carry the Control Numbers for proper correlation. Control Number is used to correlate a ACK with the sent message.

Figure 4-13 displays the available sequencing modes for outbound messages.

Figure 4-13   Outbound Message Sequencing Modes
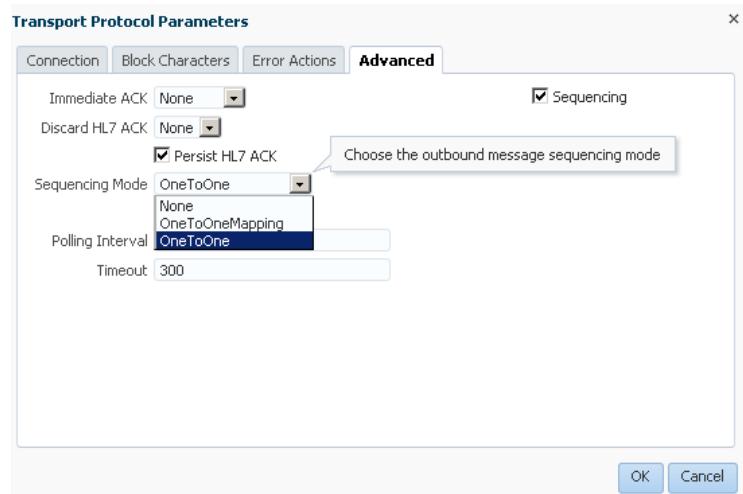
For enabling Interface Sequencing for the endpoint, select the **Interface Sequencing** check box. See Interface Sequencing for more information on configuring Interface Sequencing.

> **Note:**
>
> The value selected for the **Discard HL7 ACK** check box indicates if the ACK payload is persisted or not. This is applicable only in the case of Immediate Acknowledgements or Discard Acknowledgement cases. Deselecting the check box reduces the I/O load at the database layer due to persistence of payloads.

For more information on transport protocols and their corresponding exchange protocols, see Table 5-3 Transport Protocol Parameters and *Table 5-5 Exchange Protocol Parameters* in the Chapter 5 Configuring Trading Partners in *Oracle Fusion Middleware User's Guide for Oracle B2B*.

For a single-directional endpoint:

- When you click the **Transport Details** button, the Transport Protocol Parameters dialog box that appears, is prompts for a set of parameters that are different from the bidirectional endpoint. Figure 4-14 displays the Transport Protocol parameters dialog box for a single-directional endpoint.

Figure 4-14   The Transport Protocol Parameters Dialog Box
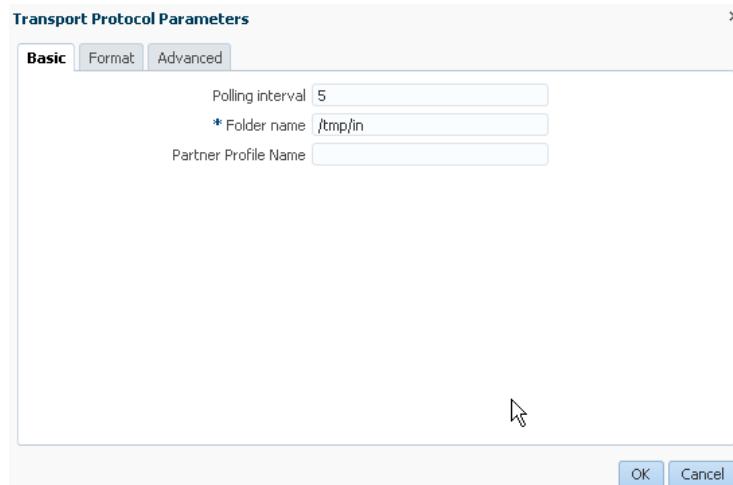
For more information on transport protocols, see Table 5-3 Transport Protocol Parameters in the Chapter 5 Configuring Trading Partners in *Oracle Fusion Middleware User's Guide for Oracle B2B*.

4.5 Enabling SSL/TLS Support for MLLP Endpoints

Oracle SOA Suite for healthcare integration provides support for exchanging messages over secured socket connections by using Secured Socket layer (SSL)/Transport Layer Security (TLS) protocol. SSL/TLS protocols provide security to Oracle SOA Suite for healthcare integration to ensure that the security of the messages are not compromised in the process of exchange by potential hackers. Oracle SOA Suite for healthcare integration provides you the option of using either SSL or TLS for securing your message exchange. Currently, this is supported only for MLLP 1.0 endpoints.

SSL is a cryptographic protocol used for transmitting private documents by using the Internet. SSL uses a cryptographic system that uses two keys to encrypt and decrypt data - a public key that is known to everyone is the network and a private key that is known only to the recipient of the message. Please see http://www.tldp.org/HOWTO/SSL-Certificates-HOWTO/x64.html to learn more about SSL.

TLS is actually a standardized version of SSL. In fact, SSL version 3 is the last update of SSL, post which TLS comes into being. Essentially, TLS is an advanced version of SSL, which provides the following benefits over SSL:

- Provides a new cryptographic encryption algorithm for securing connection

- Allows both secure and insecure connections over the same port

- Is more extensible

- Enables two-way (both server and client) authentication at the time of data exchange; server authentication is done by default

To enable SSL/TLS support for MLLP 1.0 endpoints:

1. Open the endpoint.

2. Click the **Transport Details** button to display the Transport Protocol Parameters dialog box.

3. Click the **Connection** tab to display a list of configurable connection options as shown in Figure 4-15.

Figure 4-15   Enabling SSL/TLS in an Endpoint
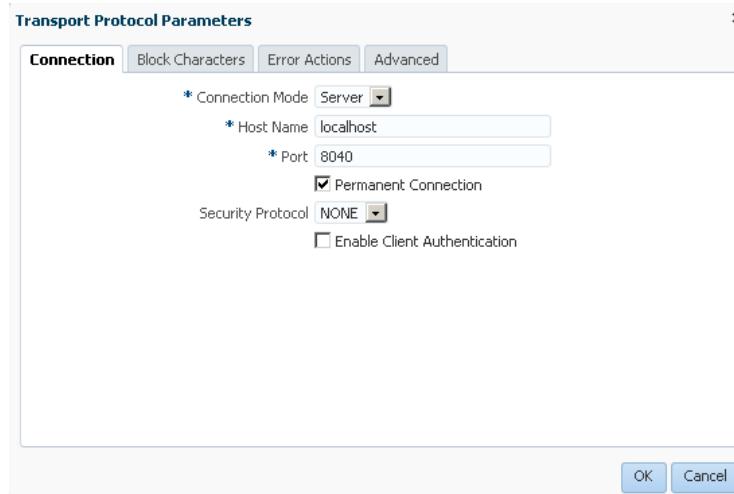
4. Select the required protocol from the **Security Protocol** list. The available options are:

- **NONE**: No security protocol to be used; this is the default selection

- **SSLv3**: SSLv3 security to be used

- **TLSv1**: TLSv1 security to be used

5. Click **Enable Client Authentication** to allow two-way authentication. By default, server certificate authentication takes place whenever a client requests for secured

data exchange. Selecting the **Enable Client Authentication** check box allows for client certificate authentication as well. However, this is optional.

Note:

The **Enable Client Authentication** check box is enabled only if you have selected **Server** as the **Connection Mode** and anything other than **NONE** as the **Security Protocol**. If you select **Client** as the **Connection Mode** and **NONE** as the **Security Protocol**, this check box is greyed out, because in case of a client endpoint, server authentication happens by default.

---

6. Click **OK** and then click **Apply** in the endpoint page.

The secured data exchange requires an infrastructure where client/server trusted certificates and the cryptographic algorithm and public/private key pairs are stored. For this, you must define keystores and specify private key passwords. For a two-way authentication, you require to configure two keystores, one at the client side to store server certificates, and the other at the server side to store client certificates.

See Table 13-1 for more information on configuring keystores and specifying private key passwords from the Oracle SOA Suite for healthcare integration console.

4.6 Handling Actionable Errors for an MLLP Endpoint

Oracle SOA Suite for healthcare integration provides support for receiving and delivering HL7 v2.x Messages over the MLLP 1.0 protocol and handling issues such as message parse failure, message validation failure, or rejection.

- *Message parse* failure is a term is used for HL7 message MSH (HL7 message Message Header Segment) parse/validation failures in SOA Healthcare.

- *Message validation failure* is a term is used for payload validation failures in SOA Healthcare.

In addition, it provides support for automatically retrying failed messages, resetting TCP connections that might be in a faulty state, and stopping message flow after several consecutive message delivery failures.

Typically, message delivery can fail due to the following errors:

- Parse failure: Caused due to an error in grammar or semantics of the HL7 message

- Validation failure: Caused to due incorrect value in an HL7 message

- ACK errors: Caused due to a negative Acknowledgement being received

In case you encounter any parse failure and validation failure errors, you can perform certain corrective actions at the server side (inbound), and in case of ACK errors, you can perform actions at the client side (outbound).

After an endpoint encounters conditions where a corrective action has to be performed (based on the preceding errors), Oracle Healthcare is capable of sending system notifications (similar to Exception messages that are sent for error encountered during processing of messages).To enable Oracle Healthcare to send system notification regarding the corrective actions taken for the preceding system errors, you must set the `hc.mllp.EnableEventNotification` server property in Oracle Fusion Middleware Enterprise Manager Control console. If this property is set to `true`, only

then Oracle Healthcare generates system notification for the system-level errors, else no system-level notification is generated.

Example - **Notification.xsd** shows the file used to send notification messages to the notification queue.

Notification.xsd

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://integration.oracle.com/B2B/Notification"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://integration.oracle.com/B2B/Notification">
  <xs:element name="Notification">
    <!--xs:complexType name="Notification"-->
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="eventType" minOccurs="0"/>
        <xs:element ref="eventCode" minOccurs="0"/>
        <xs:element ref="eventText" minOccurs="0"/>
        <xs:element ref="eventDescription" minOccurs="0"/>
        <xs:element ref="eventSeverity" minOccurs="0"/>
                <xs:element ref="ComponentDetails" minOccurs="0"/>
                <xs:element ref="eventDetails" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="eventType" type="xs:string"/>
  <xs:element name="eventCode" type="xs:string" />
  <xs:element name="eventText" type="xs:string" />
  <xs:element name="eventDescription" type="xs:string" />
  <xs:element name="eventSeverity" type="xs:string" />

  <xs:element name="ComponentDetails">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="ComponentName" minOccurs="0"/>
                <xs:element ref="ComponentType" minOccurs="0"/>
                <xs:element ref="ComponentSubType" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ComponentName" type="xs:string" />
  <xs:element name="ComponentType" type="xs:string" />
  <xs:element name="ComponentSubType" type="xs:string" />

  <xs:element name="eventDetails">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="parameter" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="parameter">
    <xs:complexType>
      <xs:attribute name="name" type="xs:string" use="required" />
      <xs:attribute name="value" type="xs:string" use="required" />
    </xs:complexType>
  </xs:element>
</xs:schema>
```

If the hc.mllp.EnableEventNotification server property is set to true, and if a Notification Queue value is specified under **Settings > Runtime> Notification Queue**

in the **Administration** tab of the Oracle Healthcare console, in the case of Reset Connection and Pause Endpoint error actions, Oracle Healthcare sends system notifications that are sent to the specified Notification Queue.

All notification messages now go to the notification queue. The user need not set the filter to differentiate between notification messages and error messages because they go to different queues. The `Message_Type` header is added to the notification messages for the following reasons:

- Backward compatibility.

- If the user does not set the notification and exception queues, both set of messages go to the same default queue. If this occurs, the Message_Type header can be used to differentiate the messages.

> **Note:**
>
> By default, the Notification Queue dropdown list is empty. To populate the list, you must use **Designer** > **Administration** > **Internal Delivery Channel** > **Send to Internal** to create a JMS channel (Send to Internal, Internal Delivery Channel).

If a Notification Queue is not specified, then Oracle Healthcare sends the system notification to the default queue or to the SOA notification composite.

You can set these actions either at:

- The endpoint level

- The global level

4.6.1 Handling Errors at the Endpoint Level

Oracle SOA Suite for healthcare integration provides you the option of handling the error messages at the endpoint level by using the Oracle SOA Suite for healthcare integration console.

To set error actions at the endpoint level:

1. Open the endpoint.

2. Click the **Transport Details** button to display the Transport Protocol Parameters dialog box.

3. Click the **Error Actions** tab to display the configurable error actions and when these actions must be performed as shown in Figure 4-16.

Figure 4-16   The Error Actions Tab
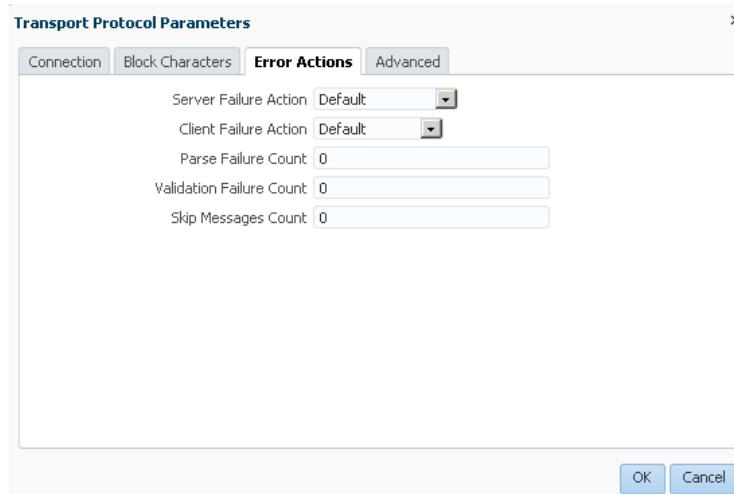
4. Select any of the actions listed in Table 4-1:

Table 4-1   Error Actions

Error Actions	Description
Server Failure Action	It is the action to be performed while receiving HL7 messages in case of any message failure, such as message parsing failure and message validation failure.
	The available options are :
	• **Default:** The property set in the Oracle Fusion Middleware Enterprise Manager Control console will be enforced. If nothing is specified, no action will be taken.
	• **None:** No action will be taken for the endpoint.
	• **ResetConnection:** The connection is reset when either the parse failure or a validation failure count reaches the specified limit. The client needs to establish the connection to send the messages again.
	• **PauseEndpoint:** The endpoint is paused when either the parse failure or validation failure count reaches the specified limit. If the soa server restarts, the endpoint will get started, however, the outbound direction will remain paused because of the entry into sequence manager table. You need to resume the outbound direction manually by using Oracle SOA Suite for healthcare integration console or by using the command line utility for further processing of messages. Once the validation failure count is reached, for Oracle Healthcare to pause the endpoint, you need to configure a Functional ACK.
	Note: For a parse failure error action to be invoked with validation failure error action, you need to configure an Immediate ACK set to **Negative**, and for validation failure error action, you need to configure a Functional ACK; else these error actions will not be invoked.

Table 4-1   (Cont.) Error Actions

Error Actions	Description
Client Failure Action	It is the action to be performed while receiving HL7 messages in case of any negative Acknowledgement being received.
	The available options are:
	• **Default:** The property set in the Oracle Fusion Middleware Enterprise Manager Control console will be enforced. If nothing is specified, no action will be taken.
	• **None:** No action will be taken for the endpoint.
	• **SkipMessage:** The message is to be skipped in case of a negative Acknowledgement being received. If the SkipMessage error action is selected, then Oracle Healthcare skips the message that has received negative acknowledgement and moves on to deliver the next message until the skip count is not reached. Once the skip count is reached, if the next message receives a negative Acknowledgement, the endpoint is paused.
	The SkipMessage action works only if the server property `b2b.discardACKList` is set to either `AE` (Application Error) or `AR` (Application Reject) or `ALL`.
	• **PauseEndpoint:** If the PauseEndpoint error action is selected, then Oracle Healthcare pauses the endpoint once a negative acknowledgment is received. The endpoint is also paused when the number of skipped messages reaches the specified limit. You need to manually resume the endpoint from the Oracle SOA Suite for healthcare integration dashboard to further process the messages.
	Note: When the client endpoint is paused, a server restart keeps the client endpoint in paused state because of the presence of the sequence manager entry, which blocks outbound messages.
	To reset the connection when the Functional ACK times out and the retries are exhausted, you need to set the server property `hc.mllp.ResetConnectionOnAckTimeOut` to `true`. and set the **Retry Interval** value under Channel/Documents. So, when the retry interval time exceeds, and if the preceding proerty is set, the connection is reset.
Parse Failure Count	The maximum number of parse failures for an endpoint before the Server Failure Action is performed.
Validation Failure Count	The maximum number of validation failures for an endpoint before the Server Failure Action is performed.
Skip Messages Count	The maximum number of consecutive messages that will be skipped before endpoint gets paused.

5. Click **OK** then click the **Save Changes** button.

4.6.2 Handling Errors at the Global Level

Oracle SOA Suite for healthcare integration provides you the option of handling the error messages at the global level by setting the Server properties by using the Oracle Fusion Middleware Enterprise Manager Control console.

To set the properties for setting error actions:

1. Log on to Oracle Fusion Middleware Enterprise Manager Control console by accessing the following:

   http://*<hostname>*:*<port>*/em, where *hostname* is the name of the computer where the Oracle Weblogic server is running, and *port* is typically 7001.

2. Navigate to **B2B Server Properties** by expanding **SOA > SOA Administration**, and then right-clicking *<soa domain name>* as shown in Figure 4-17.

Figure 4-17   Accessing B2B Server Properties
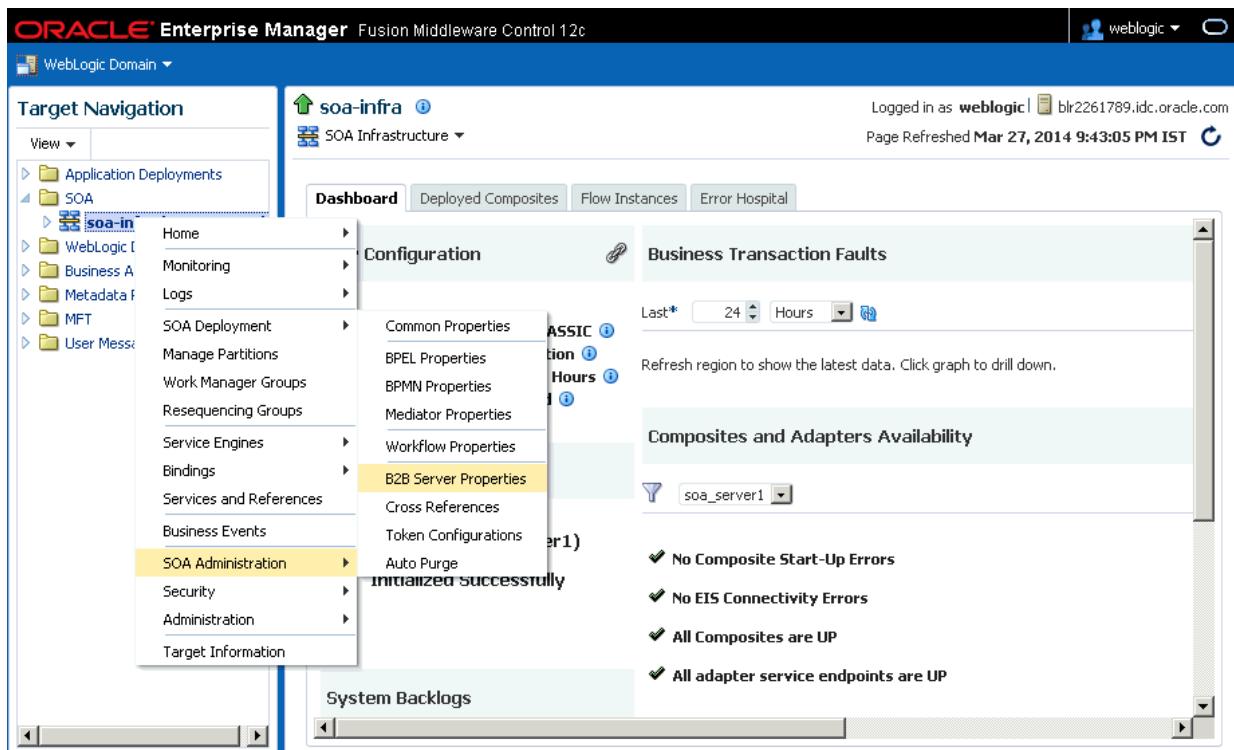
3. On the B2B Server Properties page, click the **More B2B Configuration Properties...** link to display the System MBean Browser.

4. Click the **Operations** tab and then click **addProperty** as shown in Figure 4-18.

Figure 4-18   Adding B2B Server Properties
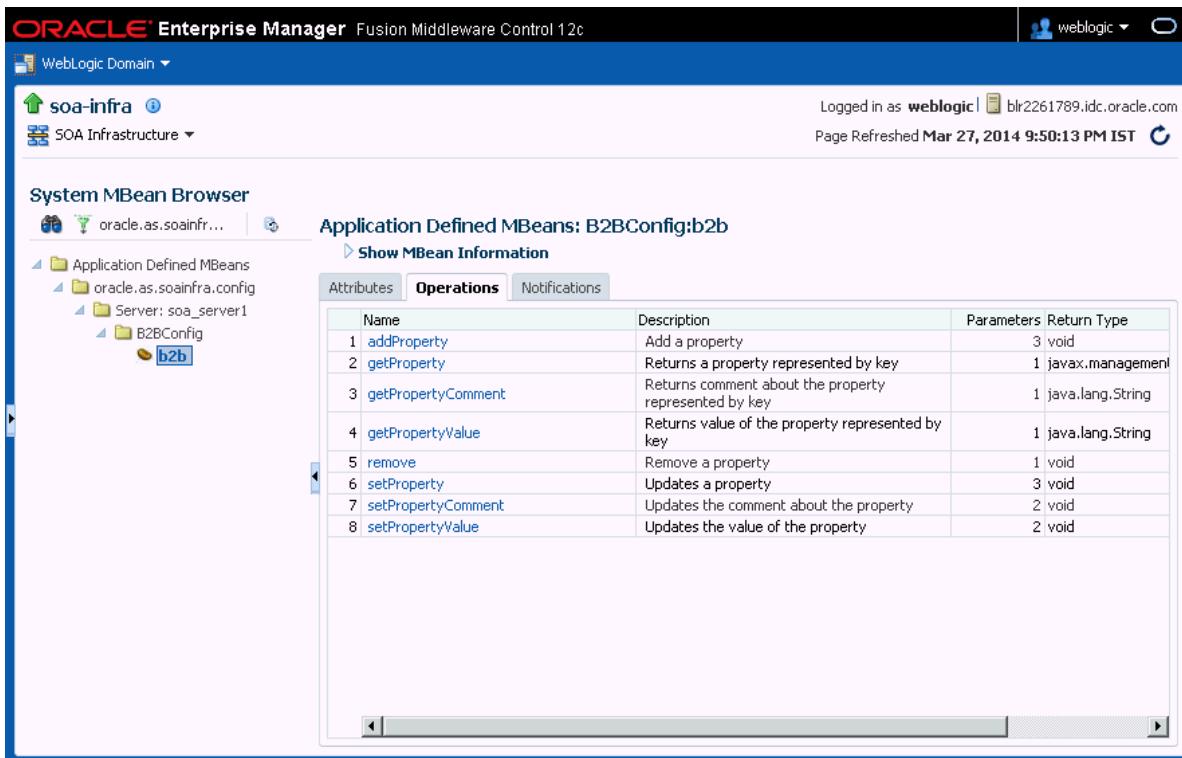
5. Add the properties listed in Table 4-2. The properties are equivalent to the error actions listed in Table 4-1 that you can configure in the Oracle SOA Suite for healthcare integration console.

Table 4-2   Server Properties for Configuring Error Actions

Property Name (`java.lang.String`)	Valid Property Values (`java.lang.String`)
`hc.mllp.ServerFailureAction`	`ResetConnection`, `PauseEndpoint`
`hc.mllp.ClientFailureAction`	`SkipMessage`, `PauseEndpoint`
`hc.mllp.ParseFailureCount`	Value greater than zero
`hc.mllp.ValidationFailureCount`	Value greater than zero
`hc.mllp.SkipMessagesCount`	Value greater than zero

The parameters set in the Oracle SOA Suite for healthcare integration console are applicable at each endpoint level, and the Server properties set using the Oracle Fusion Middleware Enterprise Manager Control console are applicable for all the endpoints. At runtime, for each endpoint, the values set at the endpoint level in the Oracle SOA Suite for healthcare integration console (other than `Default` for Failure Actions and `0` for the counts) override the values set at global level, which means that the global level values are only used if the `Default` value or `0` is selected at the console level.

4.7 Deleting an Endpoint

To delete an endpoint, you must first disable it. An enabled endpoint cannot be deleted.

To delete an endpoint, log on to the Oracle SOA Suite for healthcare integration user interface. In the **Configuration** tab under the **Designer** link, select the endpoint name and click the **Delete** button. You can also right-click the endpoint name and click **Delete** from the shortcut menu.

4.8 Working with the Endpoint Window

You can use open the Endpoint window to view a list of existing endpoints. In addition, using the endpoint window, you can create, modify or delete endpoints. You can also perform bulk modification or deletion using this window by selecting all or the required endpoints and then performing the operation of your choice.

To access the Endpoint window:

1. Log on to the Oracle SOA Suite for healthcare integration user interface.

2. In the **Configuration** tab under the **Designer** link, double-click the Endpoint folder.

displays the Endpoint window.

Figure 4-19   The Endpoint Window

4.9 Healthcare and Oracle Managed File Transfer Integration

Oracle SOA Suite for healthcare integration (Oracle Healthcare) recognizes Oracle Managed File Transfer (MFT) as a remote endpoint. In endpoint configurations, MFT

is added as additional transport protocol. Oracle Healthcare uses an outbound endpoint to send files to MFT and an inbound endpoint to receive files.

You must configure an Oracle Healthcare domain in MFT if Oracle Healthcare is not co-located with MFT. For more information, see "Managing Domains" in *Oracle Fusion Middleware Using MFT.*

4.9.1 Oracle Healthcare Endpoint Configurations

The following sections describe how to create Oracle Healthcare endpoint configurations.

4.9.1.1 Creating an Outbound Endpoint for an Oracle Healthcare Source

An Oracle Healthcare source in MFT corresponds to an outbound endpoint for a remote endpoint in Oracle Healthcare.

The steps for this process are:

1. In Oracle Healthcare, in the **Configuration** tab under the **Design** tab, click the Endpoint folder and then click the **Create** button

2. In the **Create Endpoint** dialog (Figure 4-20), enter the following and click **OK**:

   - **Name**: Type a name for the new endpoint.

   - **Transport Protocol**: Select **MFT**

   - **Direction**: Select **Outbound**

   - **Source**: Type the MFT source name

Figure 4-20   Create Endpoint Dialog
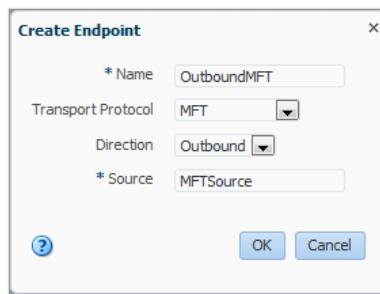
The endpoint is displayed in the right panel of the Oracle SOA Suite for healthcare integration user interface.

3. Click the **Transport Details** button.

4. In the **Transport Protocol Parameters** window (Figure 4-21), enter the following and click **OK**:

   - **Source**: Type the MFT source name.

   - **URL**: Type the URL of the MFT source.

   - **Username**: Type the MFT user name.

   - **Password, Confirm Password**: Type the password for the MFT user name

Figure 4-21   Transport Protocol Parameters Dialog
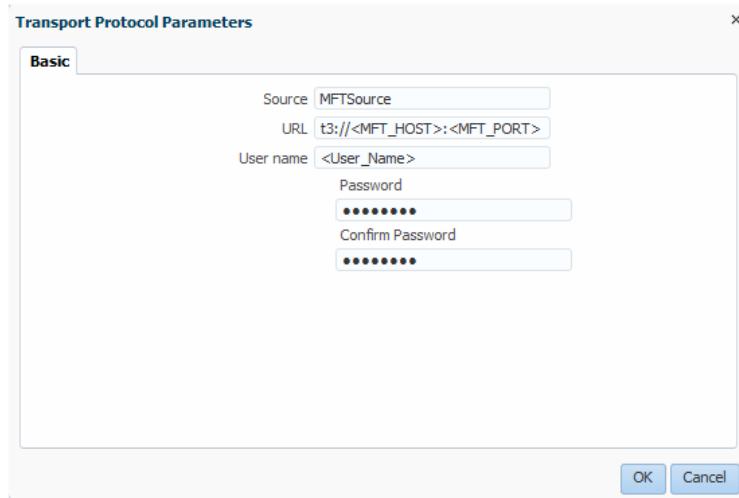
5. Find the document to be transferred under the **Document Protocol** folder

6. Drag and drop the document to be transferred into the **Document to Send** table.

7. Save the endpoint.

For more information, see the previous sections in this chapter.

4.9.1.2 Creating an Inbound Endpoint for an Oracle Healthcare Target

An Oracle Healthcare target in MFT corresponds to an inbound endpoint for a remote endpoint in Oracle Healthcare.

The steps for this process are:

1. In Oracle Healthcare, in the **Configuration** tab under the **Design** tab, click the **Endpoint** folder and then click the **Create** button.

2. In the **Create** window (Figure 4-22), enter the following and click **OK**:

* **Name**: Type a name for the new endpoint.

* **Transport Protocol**: Select **MFT**.

* **Direction**: Select **Inbound**.

Figure 4-22   Create Endpoing Dialog
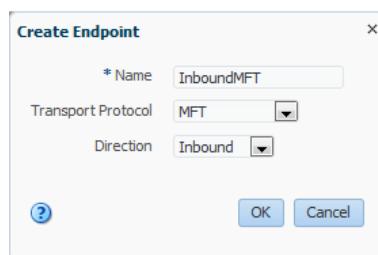
3. Find the document to be transferred under the **Document Protocol** folder.

4. Drag and drop the document to be transferred into the **Document to Receive** table.

5. Save the endpoint.

For more information, see the previous sections in this chapter.

4.9.2 Creating the Required MFT Artifacts for Oracle Healthcare

The following sections describe how to create the artifacts required for Oracle Healthcare.

4.9.2.1 Creating an MFT Domain to interact With Oracle Healthcare Endpoints

Use the following steps to configure a domain for Oracle Healthcare in the MFT user interface.

1. Open the **Domains** tab on the **Administration** page.

2. Click the **Add** button.

An empty row is added to the domain table.

3. Enter the following information in the table cells (Figure 4-23):

- **Domain Alias**: The host name used to connect to the domain. The domain alias setting for a source or target maps to this alias.

- **Connection URL**: The service endpoint URL for connecting to Oracle Healthcare applications running on remote servers. These servers can be in the same Oracle WebLogic Server domain as MFT or in a different domain. It is used to send messages to Oracle Healthcare endpoints.

- **User**: The user accessing the domain.

- **Password**: The user password.

- **Confirm Password**: User password confirmation.

- **Tracking URL**: The Oracle Healthcare URL used by MFT for constructing the Oracle Healthcare reports URL.

- **Type**: Select the domain type **Healthcare** from the dropdown list.

- **Description**: A text description of the domain.

4. Click **Save**.

Figure 4-23   Domains Table

4.9.2.2 Creating an MFT Source for an Outbound Oracle Healthcare Endpoint

An MFT outbound endpoint in Oracle Healthcare corresponds to an MFT Source in MFT.

The following steps describe how to create an MFT Source:

1. In the MFT user interface, in the **Designer** tab under the **Design** tab, click the **Sources** folder and then click the **Create** button.

2. In the **Create Source** (Figure 4-24) window, enter the following and click **OK**:

   - **Name**: Name of the new Source.

   - **Description**: Description of the new Source.

   - **Type**: Select **Source Type Healthcare** from the dropdown list.

   - **Endpoint Name**: Type the corresponding endpoint name intended for this source.

   - **Domain Alias**: Select the corresponding Oracle Healthcare domain from the dropdown list

Figure 4-24   Create Source Dialog
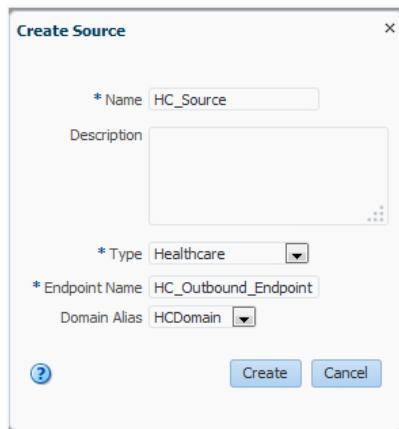
The created source is displayed in the right panel of the MFT user interface.

4.9.2.3 Creating an MFT Target for an Inbound Oracle Healthcare Endpoint

An MFT *target* corresponds to a Oracle Healthcare inbound endpoint.

Use the following steps to create an MFT target:

1. Select **Targets** in the left pane navigator and click the **Create** button

2. In the Create Target dialog (Figure 4-25), enter the following and click **OK**

   - **Name**: Name of the target

   - **Description**: Description for the new target

   - **Type**: Select **Target Type Healthcare** from the dropdown list

   - **Endpoint Name**: Type the corresponding endpoint name intended for this target.

   - **Domain Alias**: Select the corresponding Oracle Healthcare domain from the dropdown list.

Figure 4-25   Create Target Dialog
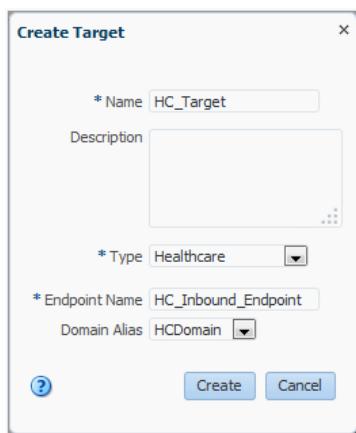
The new target is displayed in the right panel of the MFT user interface.

4.9.2.4 Creating an MFT Transfer

An MFT *transfer* is a binding entity of sources and targets created for Oracle Healthcare endpoints.

Use the following steps to create a transfer:

1. Click **Transfers** in the left pane navigator:

2. Type a **Name** and **Description** for the transfer and click **OK**. A tab for the transfer opens.

3. Add the corresponding source and targets created for Oracle Healthcare endpoints.

4. Add preprocessing and postprocessing actions such as compression and encryption.

   This is optional and applies only to targets. Source actions are added directly in the source artifact.

5. Click the **Save** button.

6. Click the **Deploy** button after saving. You can add an optional comment.

If the associated source and target have not been previously deployed, deploying the transfer automatically deploys the associated source and target.

Figure 4-26   MFT Transfer Page

4.9.3 Interlinked Oracle Healthcare and MFT Reports

An MFT instance report for a Oracle Healthcare source or target has a **Correlation Flow ID** link. Click this link to open the corresponding report for the Oracle Healthcare endpoint.

Likewise, a Oracle Healthcare endpoint instance report for MFT has an MFT source or target button. Click this button to open the corresponding MFT instance report.

See "Interpreting Source, Transfer, and Target Reports" in Oracle Fusion Middleware Using Oracle Managed File Transfer for more information about MFT instance reports.

5

Working with Callouts

This chapter describes how to create and use Java callouts in Oracle SOA Suite for healthcare integration. Oracle SOA Suite for healthcare integration provides most functionality required to build healthcare composites using various configurations available in the Oracle Healthcare console. However, you can use Java callouts to extend configurations, or add additional functionality written in Java. For example, you can use Java callout to programmatically handle non-standard delimiters in HL7 messages. Java callouts can also be used to write PGP Encryption/Decryption logic.

This chapter contains the following topics:

- Introduction to Callouts
- Types of Callouts
- Creating a Callout
- Including a Callout in an Endpoint

5.1 Introduction to Callouts

Callouts are used in environments in which a host endpoint does not use the same message format as the remote trading partner.

In addition, callouts are used to customize header information in messages.

5.1.1 Creating a Callout Library JAR File

If the callout JAR file provided with Oracle SOA Suite for healthcare integration is not sufficient for your needs, you can create your own callout JAR file outside of Oracle SOA Suite for healthcare integration, following the standards described in the *Oracle Fusion Middleware B2B Callout Java API Reference*. You can specify the directory location of this external JAR file in the **Callout Directory** field that you can access from the **Runtime** link under **Settings** in the **Administration** tab of the Oracle SOA Suite for healthcare integration user interface as shown in Figure 5-1. It is recommended that you create an external JAR file for your callouts; do not bundle your callouts with `b2b.jar`.

Figure 5-1   Specifying the Callout Directory
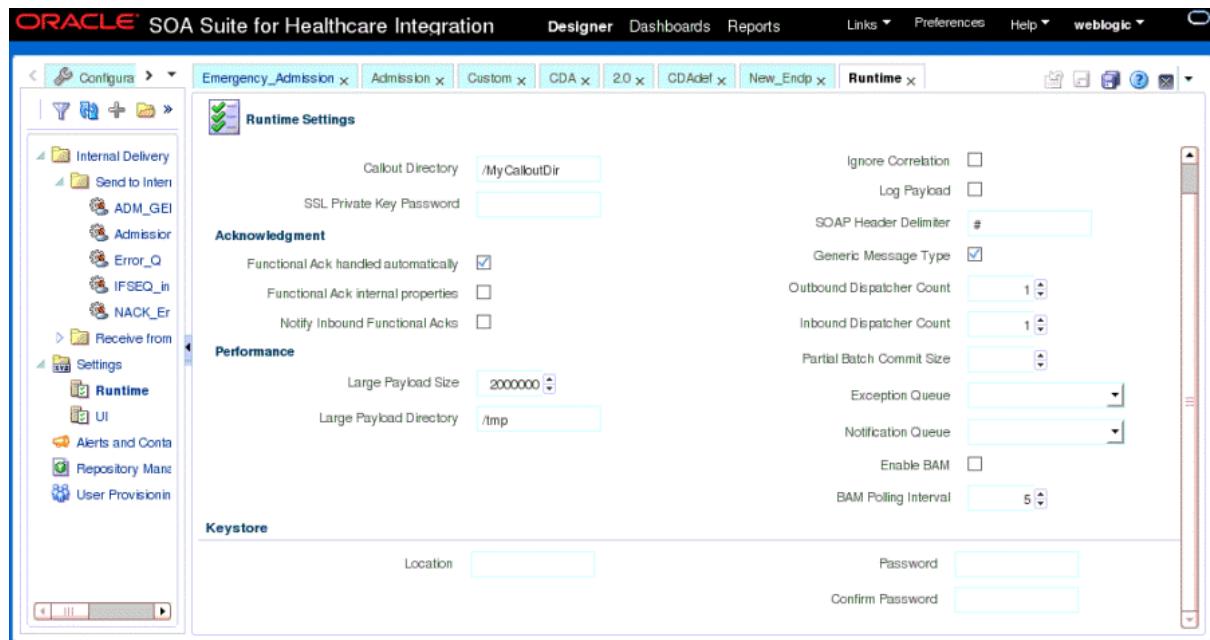
Note:

MySampleCallout is a restricted keyword and should not be used. It is already packaged into `b2b.jar`.

5.2 Types of Callouts

Oracle SOA Suite for healthcare integration provides two types of callouts:

- Transport Callouts

- Document Callouts

5.2.1 Transport Callouts

A transport callout is associated with an endpoint. For the inbound message, Oracle SOA Suite for healthcare integration invokes the transport callout immediately after it receives a message from the remote endpoint. For the outbound message, Oracle SOA Suite for healthcare integration invokes the transport callout immediately before it sends a message to the remote endpoint. Transport callouts can be selected in the endpoint configuration, as shown in Figure 5-2, and can be used with any supported protocol such as generic TCP and MLLP 1.0.

Figure 5-2   Transport Callouts
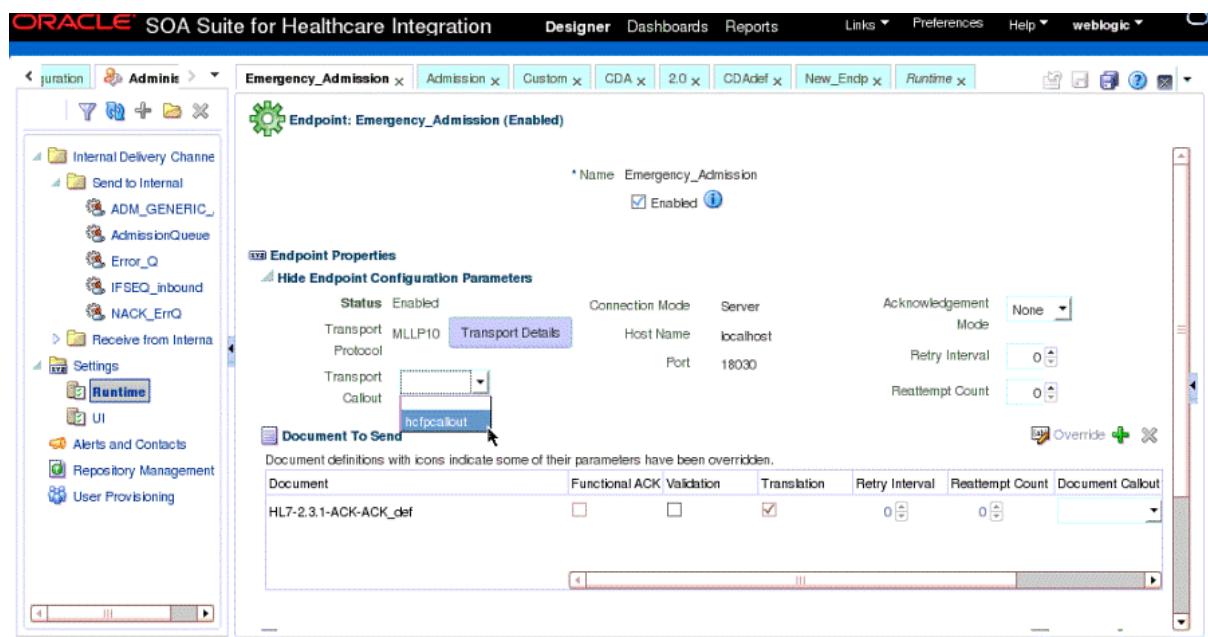
You can use transport callouts to extract transport custom headers for inbound messages and set transport headers for outbound messages. **Example - Setting and Getting the CUSTOM_HEADER Property** shows how to set and get the CUSTOM_HEADER property in the callout.

Example - Setting and Getting the CUSTOM_HEADER Property

```
import java.util.*;
import oracle.tip.b2b.callout.*;
import oracle.tip.b2b.callout.exception.*;

   public class SampleCallout implements Callout {
   public void execute(CalloutContext context,List input,List output)
             throws CalloutDomainException, CalloutSystemException {
     try {
      CalloutMessage cmIn = (CalloutMessage)input.get(0);
      String s =cmIn.getBodyAsString();

      //for getting the CUSTOM_HEADER
      Properties params =  (Properties)cmIn.getParameters();
      String customHeader = (String)params.get("CUSTOM_HEADER");

      //for setting the CUSTOM_HEADER
      CalloutMessage cmOut = new CalloutMessage(s);
      cmOut.setParameter("CUSTOM_HEADER", "your_value");
      output.add(cmOut);

      } catch (Exception e) {
      throw new CalloutDomainException(e);
      }
   }
}
```

Transport callouts can also be used for PGP encryption and decryption. The Java callout can encrypt HL7 messages being sent to an external system or endpoint. On the receiver side, a Java callout can decrypt the received data.

See "Using a Transport Callout to Extract Custom Headers" in *Oracle Fusion Middleware User's Guide for Oracle B2B* for more information.

Transport callouts are created from the **Configuration** tab under **Designer**, as described in Creating a Callout. All transport callouts appear both in the **Transport Callout** list and in the **Document Callout** list in an endpoint page; therefore, it is available for selection. To avoid confusion, when you create a transport callout, provide a name that indicates its type so that you do not select it from the **Document Callout** list.

5.2.2 Document Callouts

Document callouts are used to enable communication between endpoints that do not use the same message format. For example, a remote endpoint sends a HL7 2.3.1 XML-formatted message to a host endpoint. The host endpoint uses HL7 2.5 XML-formatted messages.

To enable communication between these two different formats, you create two callouts, as follows:

- One callout, `callout_inbound`, for example, transforms the remote message into a format understood by the host endpoint. The host endpoint, in turn, responds to the request message with a order acceptance message in HL7 2.5 XML format.

- The other callout, `callout_outbound`, for example, transforms the HL7 2.5 XML format back into an HL7 2.3.1 XML-formatted message for the remote endpoint.

These two callouts are then associated with the two endpoints, as follows:

- Associate `callout_outbound` in the endpoint for the outbound message, which is, the endpoint for the initiating message request.

- Associate `callout_inbound` in the endpoint for the inbound message, which is, the endpoint for the responding message acceptance.

Because a document definition is a component of an endpoint, a callout is associated with a specific document definition.

This example depicts a simple association of one callout to one endpoint. In reality, however, the same callout can be included in many different endpoints by changing the value of one or more callout parameters. See Figure 5-4 for where you add parameters and see Table 5-2 for a list of parameter attributes.

Document callouts can also be used to handle non-standard delimiters in HL7 messages. For example, if the receiver system sends HL7 messages using a non-standard "^" as a delimiter instead of the "|" that the healthcare composite expects, you can write a Java callout that converts the non-standard delimiters to standard delimiters.

5.3 Creating a Callout

To create a callout, provide callout details—the library name and the implementation class name—and callout parameters.

A callout is shown in Figure 5-4.

> **Note:**
>
> To create a callout from an external callout library JAR file, see Creating a Callout Library JAR File.

Figure 5-3   Creating a Callout
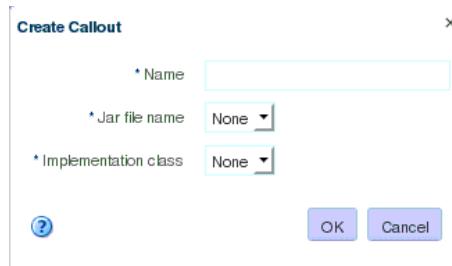
To create a callout:

1. Log on to the Oracle SOA Suite for healthcare integration user interface.

2. Click the **Designer** tab, **Configuration**, and then **Callout**.

3. Click the **Create** button (the plus sign) to display the Create Callout dialog box.

4. Enter a name for the callout.

   (You might want to indicate if you are creating a transport callout in the name.)

5. Enter callout details, as described in Table 5-1.

6. Click **OK**.

Table 5-1 lists the callout details that you provide.

Table 5-1   Callout Details

Field	Description
JAR file name	Select the library name that contains the callout implementation classes.
	Note: If you specify one or more of your own callout JAR files, you must specify the directory location. Use the **Runtime** link under **Settings** from the **Administration** link. The directory location for the default b2b.jar file included with Oracle Healthcare need not be specified.
	The callout library must be manually migrated from one environment to another. The Oracle SOA Suite for healthcare integration export/import feature does not migrate the callout library JAR.
Implementation Class	Select the implementation class name.
	Note: Oracle SOA Suite for healthcare integration includes a predefined class file named XSLTCalloutImpl that you can use for XML-to-XML transformations.

> **Note:**
>
> You cannot delete a callout that is included in an endpoint.

After creating the callout, you can configure it by specifying the time-out value and optional parameters as shown in Figure 5-4.

Figure 5-4   Configuring Callouts
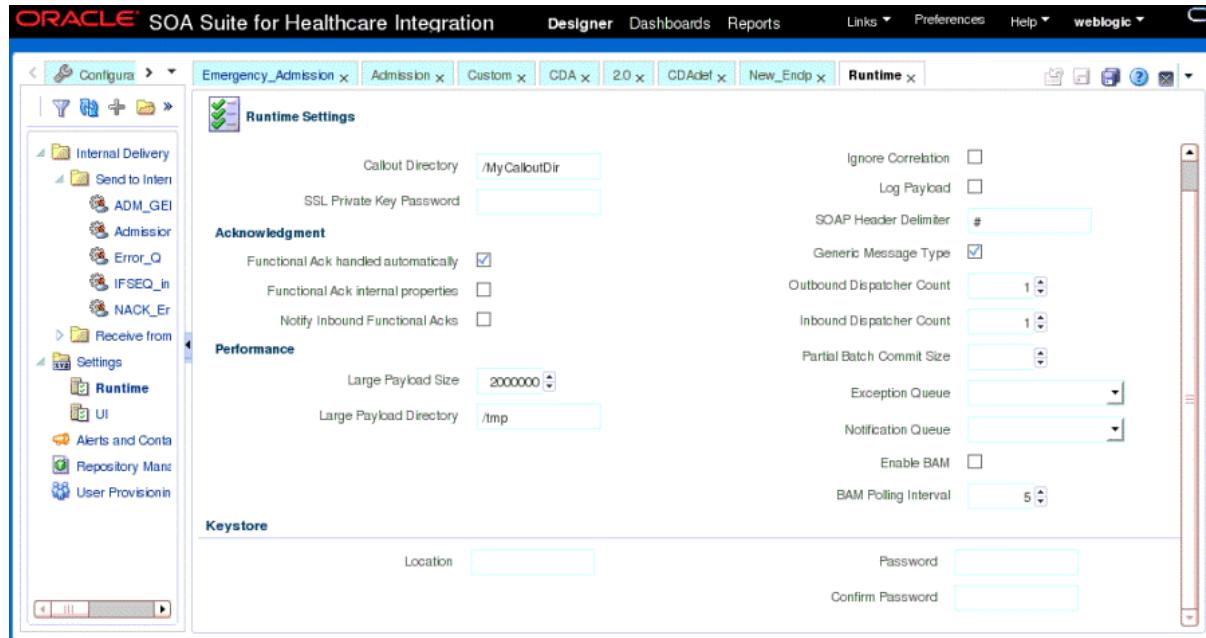
Callout parameters are similar in concept to global variables to which you can assign local values that are applicable only to a specific callout use. Or, you can create a callout parameter and assign it a default value that is applicable to all callout uses. Changes to callout parameters for an existing callout affect all endpoints that use that callout.

Table 5-2 lists the optional callout parameter attributes.

Table 5-2   Callout Parameter Attributes

Field	Description
Name	Enter a parameter name.
Type	Select from **Integer**, **Float**, **String**, **Boolean**, or **Date** types. The format for the **Dat**e type is MM/DD/YYYY. **Note:** Changing a type can invalidate the parameter default value.
Value	Enter a value. If **Encrypted** is set to **True**, then this value is encrypted.
Mandatory	Select **True** or **False**.
Encrypted	Select **True** or **False**.

Table 5-2   (Cont.) Callout Parameter Attributes

Field	Description
Description	Enter an optional description.

After you create a callout, it is available to include in an endpoint. See Including a Callout in an Endpoint, for more information. If you change a callout after it is deployed with an endpoint, a server restart is required.

5.4 Including a Callout in an Endpoint

After you create a callout, it is available to include in an endpoint.

To include a callout in an endpoint:

1. From the Configuration tab, double-click an endpoint to open it.

2. Depending on the type of message (inbound or outbound), select the callout name from the **Document Callout** list under the **Send** or **Receive** section. as shown in Figure 5-5.

Figure 5-5   Associating a Callout with an Endpoint

3. Click **Apply**.

6
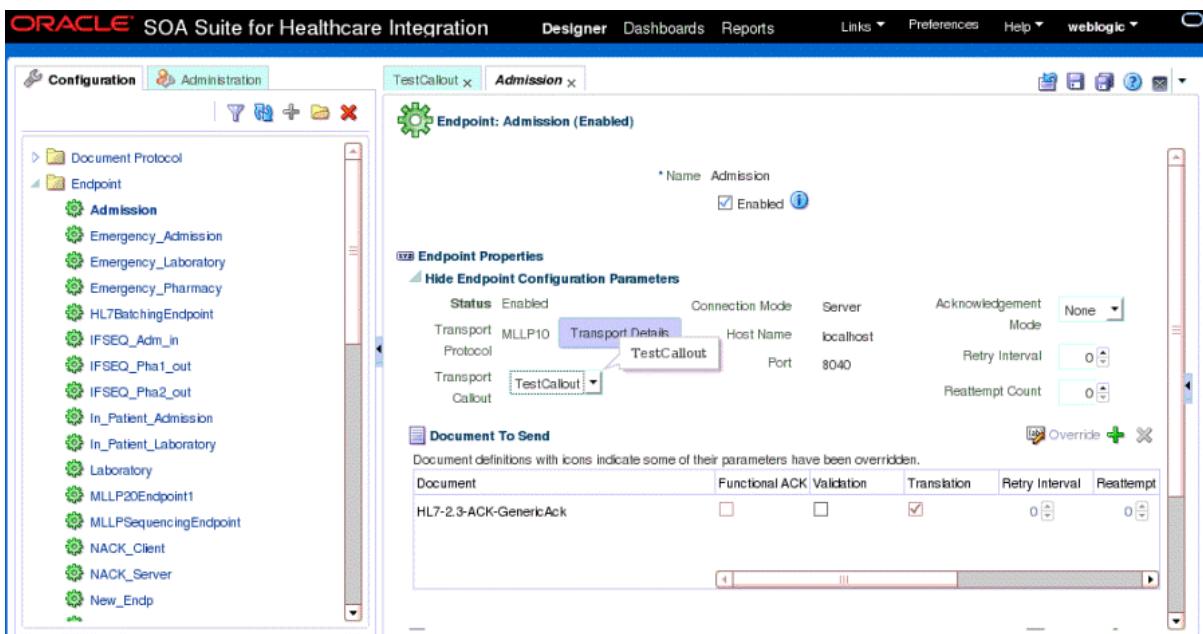
Working with Mapsets

This chapter describes how to work with mapsets in Oracle SOA Suite for healthcare integration. When the messages in your source and target systems are defined by different document definitions and you must map the data from one message to another, you can use a mapset to define the mapping logic.

This chapter includes the following topics:

- Introduction to Mapsets

- Creating a Map File

- Using Mapsets in Oracle SOA Suite for Healthcare Integration

6.1 Introduction to Mapsets

Mapsets provide data transformation for cases where it is better to map a native data format to a native data format instead of the standard translation of mapping native formatting to XML, XML to XML, and then XML back to native formatting.

You can use mapsets when you must map data between messages that are defined by different document definitions. For example, you might have a system that sends messages in HL7 2.3.1 format, but a receiving system requires the data in HL7 2.5 format. Or you might have to transform HIPAA 4010 messages to HIPAA 5010 and back again.

A mapset includes a predefined or user-defined map file and two document definitions. The map file defines how data is converted from one format to the other. Currently, the Oracle SOA Suite healthcare integration user interface supports mapping different versions of HL7 messages to each other. The Oracle B2B Console supports mappings for different HIPAA X12 messages to each other, as well as different versions of HL7 to each other.

> **Note:**
>
> Currently, using mapsets to transform HL7 2.x messages to HL7 v3.0 is not supported, but this can be achieved using the XSLT mapper in Oracle JDeveloper.

6.1.1 About Mapsets

When you create a mapset, you associate a map file, a source document definition, and target document definition within one mapset. A mapset groups the definitions and mapping together into one unit that can be reused in multiple Oracle B2B or healthcare integration applications. Each mapset uses two different document definitions, and these definitions must be created in the design-time repository in order to create the mapset. Mapsets give you the option of using the default document

definitions provided for each protocol or customized definitions that you have created or modified.

When you create a mapset in Oracle SOA Suite for healthcare integration, you associate it with the appropriate endpoints to incorporate the mapping logic into healthcare integration projects. When you create a mapset in Oracle B2B, you associate it with trading partner agreements. When you associate a mapset with an endpoint or agreement, you can only select an mapset whose document definitions and message flow match that of the endpoint or agreement.

6.1.2 Predefined and Custom Mapsets

In Oracle SOA Suite for healthcare integration, you have the option of creating your own custom mapsets using the MapBuilder feature of the Oracle Document Editor or purchasing predefined map files provided by Edifecs. The prebuilt maps include maps for the Health Insurance Portability and Accountability Act (HIPAA); for example, to transform HIPAA 4010 messages to HIPAA 5010 messages.

6.2 Creating a Map File

Before you can create a mapset in either the Oracle SOA Suite for healthcare integration user interface or the Oracle B2B Console, you must have a map file that defines the mapping between the two types of document definitions.

Edifecs provides some predefined map files that you can use, or you can create the files using the MapBuilder component of the Oracle Document Editor.

> **Note:**
>
> You can download the Oracle Document Editor from the installation package for Oracle SOA Suite for healthcare integration.
>
> For more information, see "Creating Guideline Files" in *Oracle Fusion Middleware User's Guide for Oracle B2B*.

6.3 Using Mapsets in Oracle SOA Suite for Healthcare Integration

Use the Oracle SOA Suite for healthcare integration user interface when you want to map different HL7 standards to one another in a healthcare integration project.

Perform the following steps to incorporate a mapset into an Oracle B2B trading partner agreement:

- Creating a Mapset in the Healthcare Integration User Interface

- Associating a Mapset with an Endpoint

- Deleting a Mapset in the Healthcare Integration User Interface

6.3.1 Creating a Mapset in the Healthcare Integration User Interface

Before you begin, make sure the map file is available on the computer from which you are accessing the healthcare integration user interface, and that the required document definitions are already created in Oracle SOA Suite for healthcare integration for both of the standards you are mapping. The mapset cannot be created without these three components.

For information about creating document definitions in Oracle SOA Suite for healthcare integration, see Working with Document Types and Protocols.

To create a mapset

1. On the Oracle SOA Suite for healthcare integration user interface, click the **Designer** tab and then click the **Configuration** tab.

2. In the navigation panel on the left, select **Mapset** and then click **Create Mapset** (the plus button).

    The Create Mapset dialog appears.

3. Enter a unique name and a brief description of the mapset, and then click **OK**.

    Figure 6-1   Create Mapset Dialog
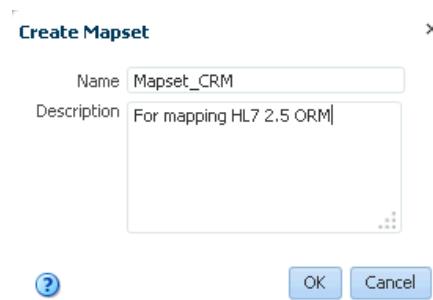
    The Mapset page appears.

4. Click **Browse** next to the **Mapset File Location** field, and then browse to and select the map file to use.

    The file is validated and if the corresponding document definitions are found, information about the document protocols, versions, and types are populated in the lower portion of the page. The default source and target definition files are automatically populated for you.

Figure 6-2   Mapset Page on the Healthcare Integration User Interface
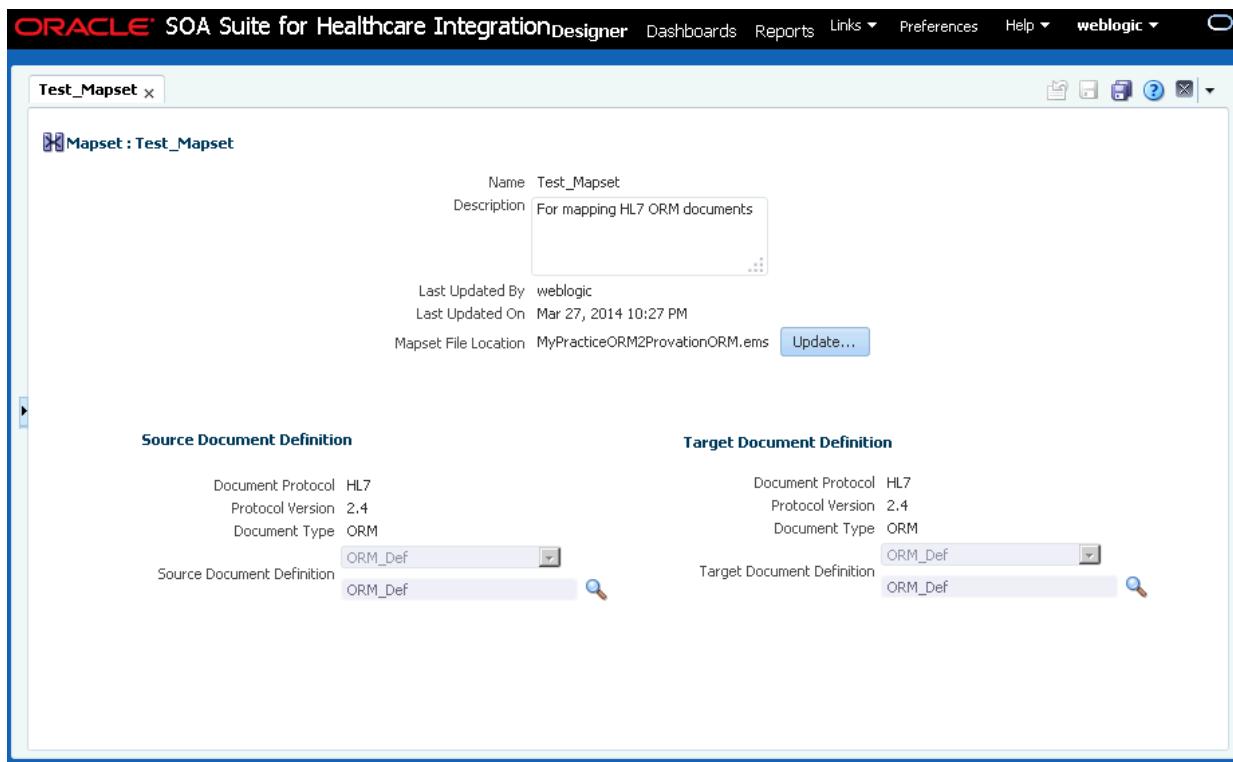
5. If you do not want to use the default version of the document definition files, do the following to override the default file selection:

    a. Next to the **Document Definition** field you want to override (Source or Target), click **Browse**.

    b. On the Document dialog, expand the Document Protocol tree until you see the document definition to use.

    c. Select the overriding document definition, and then click **OK**.

6. On the Mapset page, click **Apply** and then click **OK** on the confirmation dialog that appears.

6.3.2 Associating a Mapset with an Endpoint

After you create a mapset, you must associate it with an endpoint to include the mapping logic in the process.

To associate a mapset with an endpoint

1. On the Oracle SOA Suite for healthcare integration user interface, click the **Designer** tab and then click the **Configuration** tab.

2. In the navigation panel on the left, expand **Endpoint** and double-click the endpoint you want to associate with the mapset.

3. In the row of the document type you want to map, clear **Translation**.

> **Note:**
>
> If Translation is selected, it is ignored when used in conjunction with a mapset in the endpoint. The mapset performs the data transformation and translation is not required.

4. In the same row, click in the **Mapset** field to the far right of the table, and select the mapset to use.

Figure 6-3   Mapset Selected for an Endpoint
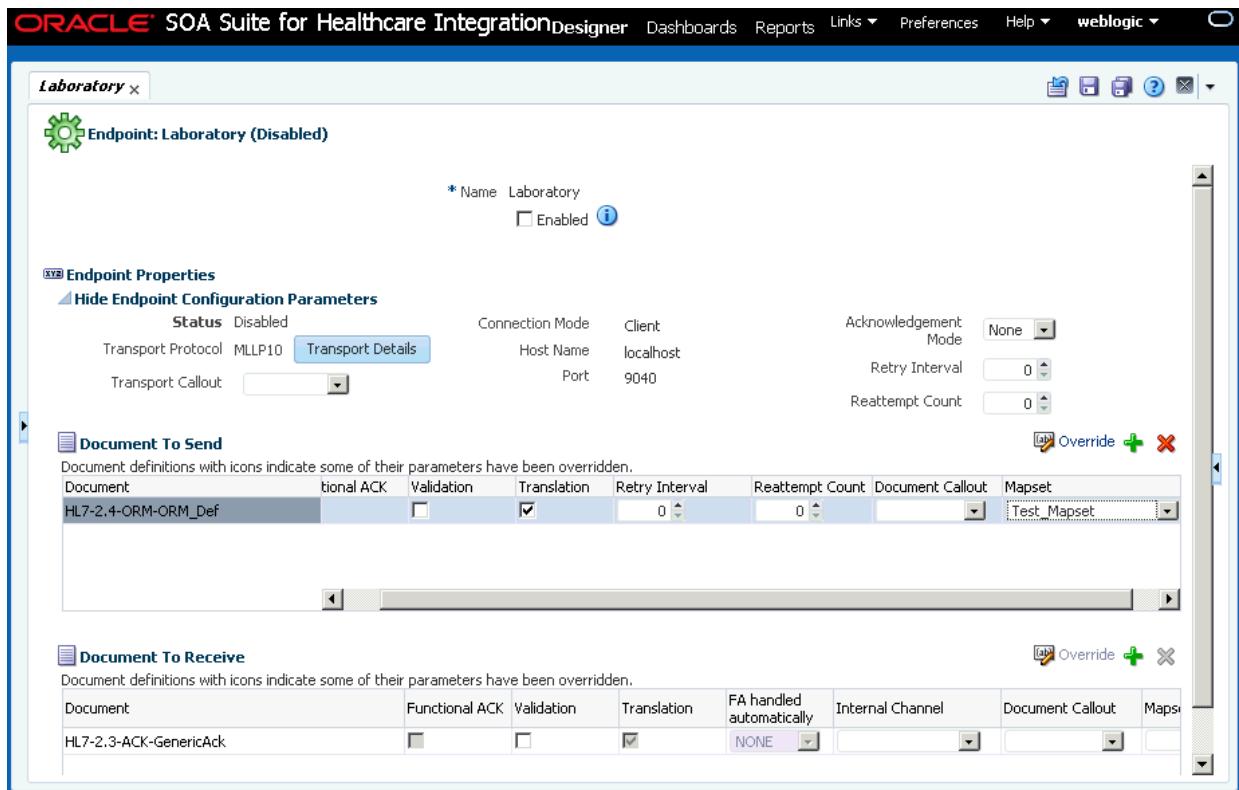
5. Click **Apply**, and then click **OK** on the confirmation dialog that appears.

6.3.3 Deleting a Mapset in the Healthcare Integration User Interface

To delete a mapset, select the mapset in the Configuration tree and click **Delete** in the toolbar. You can also right-click the mapset and then select **Delete**.

7

Working with Internal Delivery Channels

This chapter describes how to use internal delivery channels in Oracle SOA Suite for healthcare integration. Internal delivery channels are used for communicating with back-end applications, such as receiving messages from or sending them to a JMS topic or queue in a healthcare integration application.

This chapter includes the following topics:

- Introduction to Internal Delivery Channels
- Creating Internal Delivery Channels
- Enabling an Internal Delivery Channel
- Deleting an Internal Delivery Channel
- Correlating Messages Using JMS Queues

7.1 Introduction to Internal Delivery Channels

An internal delivery channel defines how a message received from an external system (endpoint) is delivered from Oracle SOA Suite for healthcare integration to back-end applications, such as JMS topics and queues, or how a message that was sent from a back-end system is received by Oracle SOA Suite for healthcare integration for delivery to an external system.

It defines the connection information, the transport protocol, acknowledgments, and so on. When you create an internal delivery channel, that channel is available to all endpoints. This avoids having to create a unique internal delivery channel for each endpoint.

7.2 Creating Internal Delivery Channels

If you must send messages to an internal delivery channel, create a "send to" internal delivery channel to associate with the endpoint. If you must receive messages from an internal delivery channel, create a "receive from" internal delivery channel.

To create an internal delivery channel

1. On the Oracle SOA Suite for healthcare integration user interface, select the Designer tab and then the Administration tab.

2. Do one of the following:

   - To create an internal delivery channel for sending messages to a JMS or queue, right-click **Send to Internal** and then click **Create**.

   - To create an internal delivery channel for receiving messages from a JMS or queue, right-click **Receive From Internal** and then click **Create**.

The Create Internal Delivery Channel dialog appears.

Figure 7-1 Create Dialog for Internal Delivery Channel (Send to)
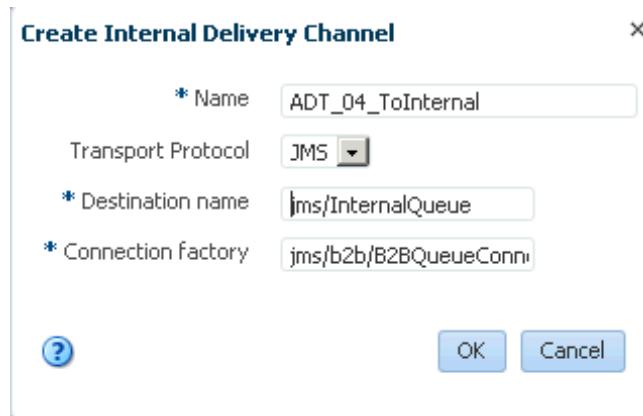
3. On the Create dialog, fill in the following fields:

Table 7-1 New Internal Delivery Channel Properties (Send to)

Property	Description
Name	A unique name for the delivery channel.
Transport Protocol	The connection protocol for the internal delivery channel. JMS is the supported protocol
Destination Name	The JNDI name of the topic or queue to which Oracle SOA Suite for healthcare integration sends messages.
Connection Factory	The JNDI location or Java class name for the connection factory, such as `jms/b2b/B2BQueueConnectionFactory`.

4. Click **OK**.

The new internal delivery channel is added to the Administration tree under Send to Internal, and the new channel appears on the Internal Delivery Channel page.

Figure 7-2   Internal Delivery Channel Page

5. On the main Internal Delivery Channel page, you can modify any of the fields listed in Table 7-1 (except Transport Protocol).

Note:

The fields in the upper portion of the Internal Delivery Channel page (**Acknowledgement Mode**, **Retry Interval**, and **Reattempt Count**), are not currently used for internal delivery channels.

---

6. To modify the transport protocol connection settings, do the following:

    a. On the main Internal Channel Delivery page, click **Transport Protocol**.

    b. On the Transport Protocol Parameters dialog, click the Basic tab, and modify any of the properties listed in Table 7-2.

Table 7-2   Internal Delivery Channel Transport Protocol Basic Parameters

Parameter	Description
Destination name	The JNDI name of the topic or queue to which Oracle SOA Suite for healthcare integration sends messages.
Connection Factory	The JNDI location or Java class name for the connection factory, such as `jms/b2b/B2BQueueConnectionFactory`.
Is Topic	An indicator of whether the destination is a topic or a queue. Select this option if the destination is a topic.
Polling Interval	The length of time in minutes between polling attempts for messages.

    c. On the Transport Protocol Parameters dialog, click the Advanced tab, and modify any of the properties listed in Table 7-3.

Table 7-3   Internal Delivery Channel Transport Protocol Advanced Parameters

Parameter	Description
Message Type	Select one of the following JMS message type options: **BYTES**, **TEXT**, or **MAP**.
Is Map Payload Alone	An indicator of whether the payload is sent alone as part of a JMS message of the type `javax.jms.MapMessage`.
Use JMS ID	An indicator of whether to use the JMS message ID as the healthcare integration message ID. This facilitates correlation at the JMS level.
Destination Provider	JNDI properties that are required to connect to the target server. Use a semicolon (;) as the separator for key/value pairs. This is for enabling Oracle SOA Suite for healthcare integration to connect to JMS queues or topics available on remote servers.
User name	The user name to connect to the target server. This value is optional for JMS because Oracle SOA Suite for healthcare integration can use the configured JNDI data sources to connect to queues.
Password (and Retype Password)	The password for the above user name.
Subscriber ID	An indicator of whether the JMS subscriber ID is required when JMS is communicating with a topic.
Sequencing	An indicator of whether messages must be delivered in sequence. Select this check box for sequential delivery. This option only applies to WebLogic Server JMS (it uses the Unit-of-Order feature of WebLogic Server JMS).

7. When you are done making changes to the transport protocol, click **OK** on the Transport Protocol Parameters dialog.

8. When you are done making changes to the internal delivery channel, click **Apply** on the Internal Delivery Channel page.

7.3 Enabling an Internal Delivery Channel

Before you can use an internal delivery channel in an Oracle SOA Suite for healthcare integration project, you must enable it.

To enable it, double-click the internal delivery channel to open the Internal Delivery Channel page, and then select the Enabled option.

7.4 Deleting an Internal Delivery Channel

To delete an internal delivery channel, select the channel in the Administration tree and click **Delete** in the toolbar.

You can also right-click the internal delivery channel and then select **Delete**.

7.5 Correlating Messages Using JMS Queues

You can correlate inbound and outbound messages using JMS queues, by setting
A2A=true in the JMS header.

If the message ID (MSG_ID) is provided from a back end application, then MSG_ID is
set to JMS Correlation ID in the healthcare integration output, otherwise the JMS
Message ID is set to JMS Correlation ID in the healthcare integration output.

8

Working with Dashboards

This chapter describes Oracle SOA Suite for healthcare integration dashboards, which display endpoint-level status and volume metrics on instance data. This includes endpoint status, message counts, sequencing information, and error messages.

This chapter includes the following topics:

- Introduction to Dashboards
- Working with System Dashboard
- Creating and Configuring Dashboards
- Viewing Information in Dashboards
- Working with Sequenced Messages
- Viewing Endpoint Error Messages

8.1 Introduction to Dashboards

Oracle SOA Suite for healthcare integration dashboards display information about the current health of the endpoints in a healthcare integration application. You can create and configure multiple dashboards as required to monitor the status and volume metrics for the endpoints you have defined.

The Dashboards tab reflects changes that occur in the runtime repository, such as purging runtime instance data, new messages processed, and new error messages. You can display data for various time periods, and you can manually refresh the data in real time or set the dashboard to automatically refresh at set intervals. The available time periods are configurable in the UI settings on the Administration tab.

When you first open a dashboard, based on the View Mode options, **Compact** and **Normal**, a summary of all the endpoints included in the dashboard appears. For each endpoint, you can view more detailed information about not only the status and message volume but also about the endpoint's configuration. Figure 8-1 shows endpoint summaries in the **Normal** mode on the Dashboard page.

Figure 8-1   Endpoint Summaries on the Dashboard
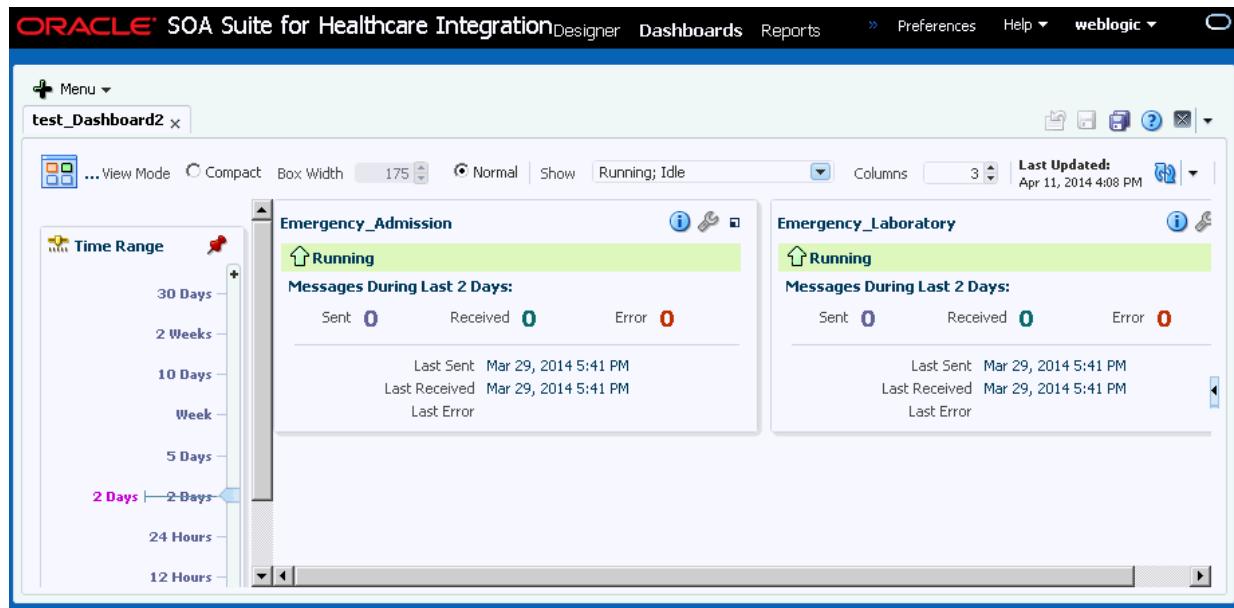
You can click the **Compact** option to view the Dashboard in the compact mode where only a small box filled with a color corresponding to the status and name of each endpoint. You can also adjust the width of the compact boxes to fit as many as you can. The Time Range can also be iconified to save space and it automatically reappears when you mouse over the Time Range slider icon. The **Show Customizer** toggle button always appears on the dashboard regardless of whether the Customizer is open or not.

Figure 8-2 shows the Dashboard page in the compact mode.

Figure 8-2   Dashboard Page - Compact Mode
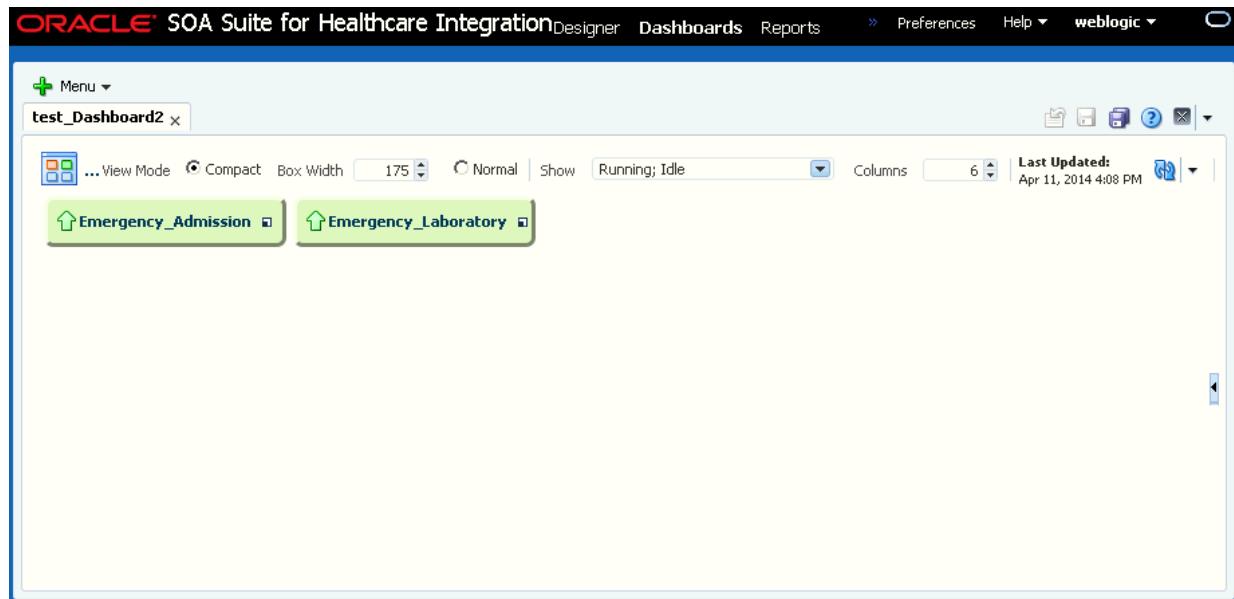
8.2 Working with System Dashboard

The Oracle Healthcare console provides a feature called the System Dashboard that displays a global view of status of the healthcare setup. This includes a overall view of

the endpoints and internal delivery channels for the purpose of monitoring. The System Dashboard enables you to view a list of enabled as well as disabled endpoints and the ports to which these endpoints listen, if applicable. In addition, the dashboard displays the message counts. The System Dashboard also allows you to group or sort endpoints by transport protocol, outbound and inbound, or both.

Using the System Dashboard, you can view the deployed healthcare composites, which is important because internal delivery channels and composites represent the internal side of the system overview. You can also enable or disable endpoints and channels by using the System dashboard. Sections on message statistics and overall recent error list completes the System Dashboard.

You can view the System Dashboard by first clicking the plus button in the upper right corner of the Dashboard tab, and then clicking **System Dashboard**.

Figure 8-3 displays a sample System Dashboard.

Figure 8-3   A Sample System Dashboard
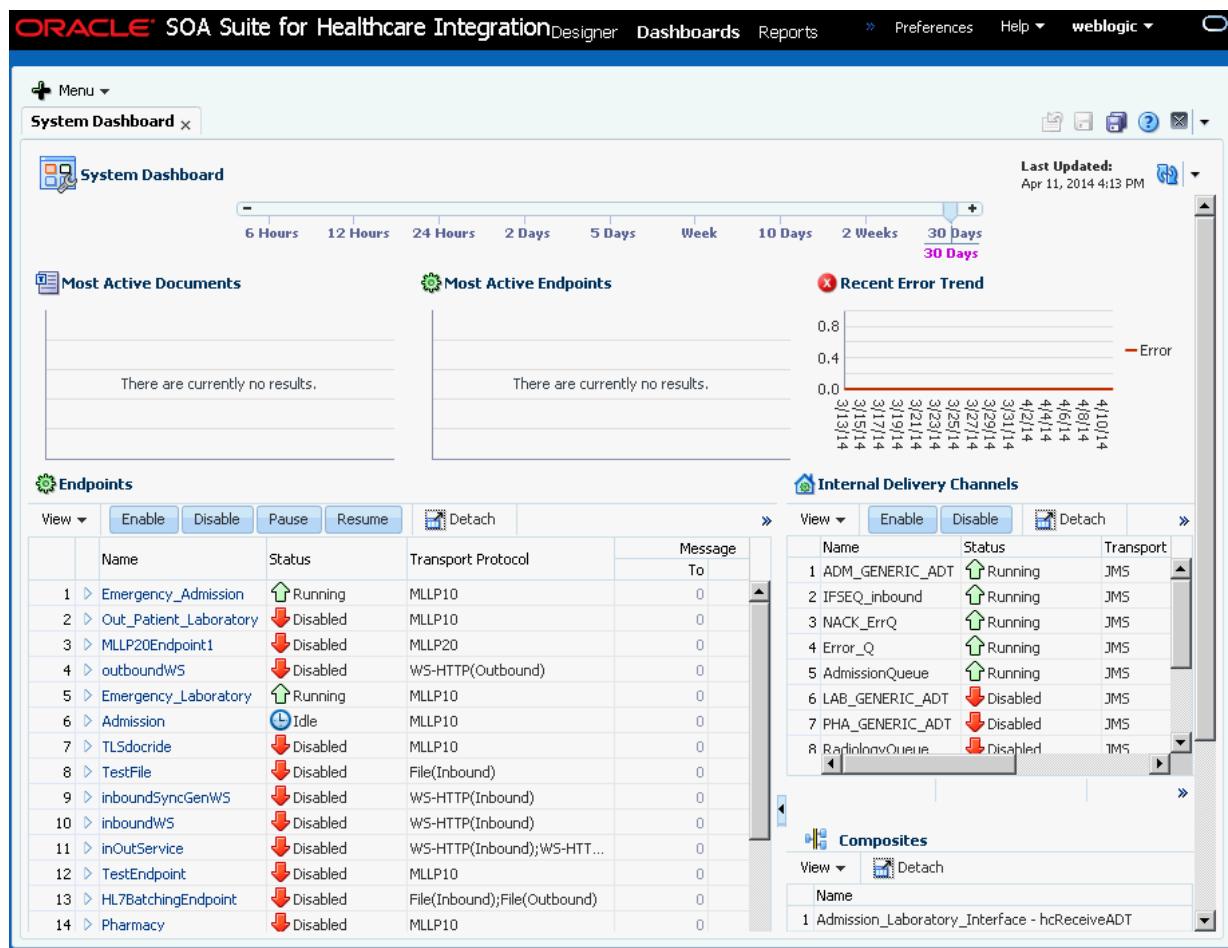
8.2.1 Viewing System Data by Using Different Sections of the System Dashboard

Using the sections of the System Dashboard, you can view different types of system data pertaining to the a Oracle Healthcare instance.

Table 8-1 lists the different sections of the System Dashboard.

Table 8-1   Sections of the System Dashboard

Section Name	Description
Slider	The time slider at top of the System Dashboard page allows you to select time intervals for which data is collected for the graphs and tables of the dashboard.
	Slide the point from to select the required time interval.
Refresh	The Refresh component at the top right corner of the dashboard allows you user to either manually refresh the page or set the interval for auto-refresh.
	To enable auto-refresh, click the down arrow adjacent to the Refresh button, select the **Auto-Refresh** check box, and specify the interval for auto-refresh (in seconds.)
Most Active Documents	This graph displays the most active document definitions (maximum of three document definitions at a time) and how many messages are associated with each document definition.
Most Active Endpoints	This graph displays the most active endpoints (maximum of three endpoints at a time) and how many messages are associated with each endpoint.
Recent Error Trend	This graph displays the number of errors in the specific Oracle Healthcare setup mapped against the time at which the errors occurred.

Table 8-1 (Cont.) Sections of the System Dashboard

Section Name	Description
Endpoints Table 	The Endpoints table displays a list of all endpoints in the Oracle Healthcare system along with the status, transport protocols, message counts, and queue counts associated with each of the endpoints. You can perform the following actions in this table: • Click the **View** button to configure the number of columns (Name, Status, Transport Protocol, and so on) that you want to show in the table (by clicking **View** > **Columns** and the required option) or detach the Endpoints table from the System Dashboard to a new window (by clicking **View** > **Detach**.) • Select multiple endpoints, and enable or disable them by clicking the **Enable** or **Disable** buttons. • Click the **Pause** button to pause the selected endpoint. • Click the **Resume** button to resume a paused endpoint. • Click the **Detach** button to detach the Endpoints table from the System Dashboard to a new window. • Filter an existing endpoint by specifying a string containing some letters of the endpoint name in the **Search** field. • Click an endpoint name to open a dashboard tab that displays the details for that endpoint. • Click the message count of either Inbound or Outbound messages to open a Report tab that displays all the relevant messages. • Sort the endpoints based on any of the column headers.
Internal Delivery Channels Table 	The Internal Delivery Channels table displays a list of all the internal delivery channels in the Oracle Healthcare system along with the status, transport protocols, and delivery associated with each of the channels. Here, the channels are listed with values `true` or `false` in the **Delivery** column differentiating them as either Internal Delivery channels (Delivery column shows `true`) or Internal Listening channels (Delivery column shows `false`). In this table, you can perform actions similar to the ones that you can perform in the Endpoints table.
Composites Table 	The Composites table displays a list of Oracle Healthcare related composites that have been deployed in the Oracle Weblogic Server along with the status of the composites Click the **View** button to configure the number of columns to be displayed in the table or to detach the Composites window .

8.3 Creating and Configuring Dashboards

Using the Dashboards tools, you can create dashboards for any combination of endpoints you have defined.

You can configure the layout of the Dashboard page, specify a refresh rate, and change the endpoints for an existing dashboard.

8.3.1 Creating a Dashboard

You can create multiple dashboards in Oracle SOA Suite for healthcare integration for a single endpoint or combinations of multiple endpoints.

Note:

Whenever there are any Endpoints defined, a standard dashboard called All Endpoints is available that includes all the defined endpoints. You cannot modify and save changes to this standard dashboard.

To create a dashboard

1. On the Oracle SOA Suite for healthcare integration user interface, click the **Dashboards** tab.

    The main Dashboards page appears.

Figure 8-4   Main Dashboards Page
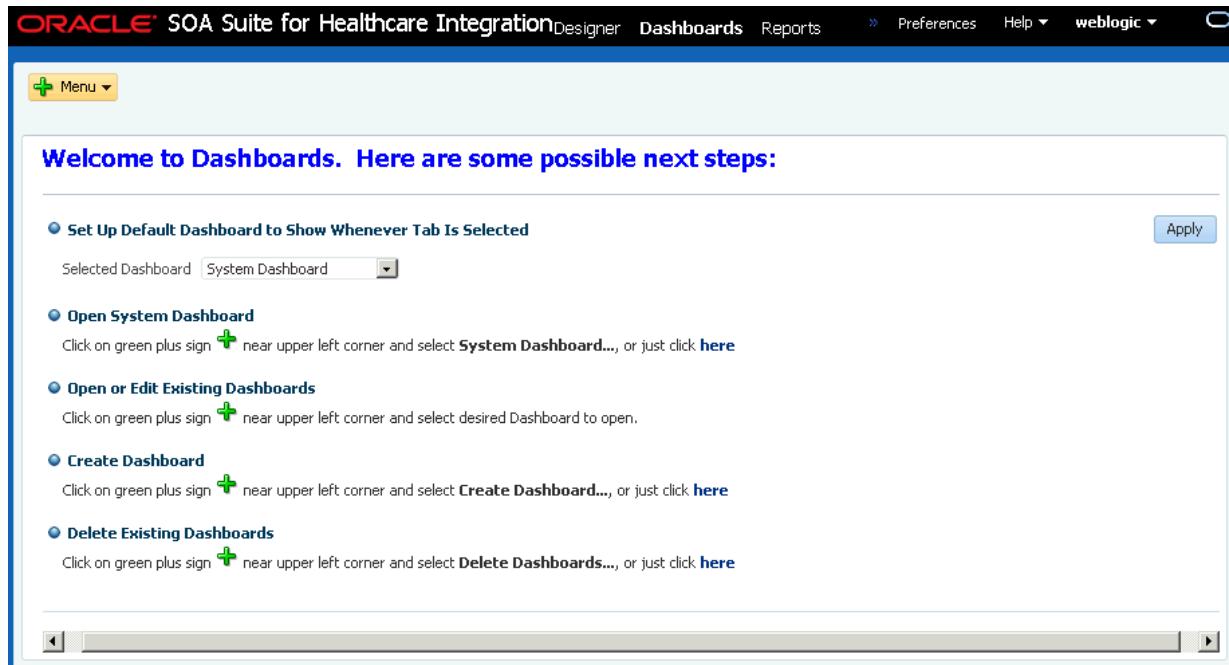
2. Click the plus button in the upper right, and select **Create Dashboard**.

    The Create Dashboard dialog appears.

Figure 8-5   Create Dashboard Dialog
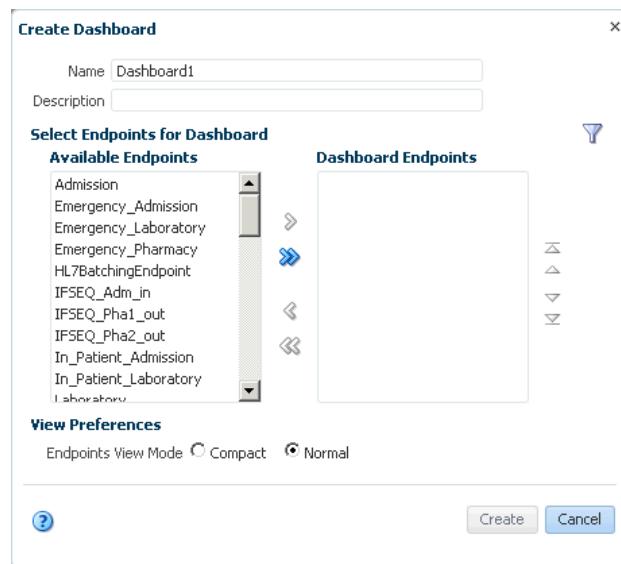
3. Enter a unique name and a description for the dashboard.

4. Under Available Endpoints, select the endpoints you want to include on the dashboard and then click the right arrow button.

The selected endpoints appear in the Dashboard Endpoints list.

Figure 8-6   Create Dashboard Dialog with Endpoints Selected
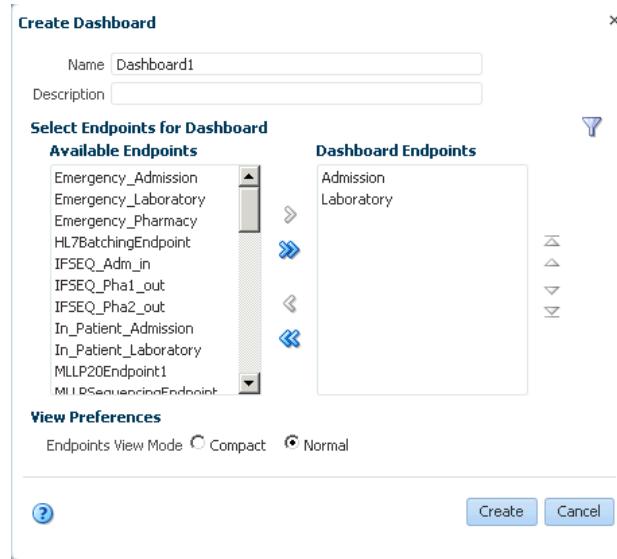
5. Select the required Endpoint View Mode.

6. Click **Create**.

The new dashboard appears with summary information displayed for each endpoint about the messages processed in the past 24 hours.

8.3.2 Selecting a Default Dashboard

After you create dashboards you can select one that automatically appears when you click the Dashboards tab. This is the default dashboard. If you do not specify a default dashboard, you can select from the list of existing dashboards when you open the Dashboards page. If you do select a default dashboard, this dashboard appears every time you open the Dashboards page.

To select a default dashboard

1. On the Oracle SOA Suite for healthcare integration user interface, click the **Dashboards** tab.

2. Click the down arrow next to the **Selected Dashboard** field, and select the dashboard that you want to make default.

Figure 8-7   Default Dashboard Selected on the Main Dashboards Page
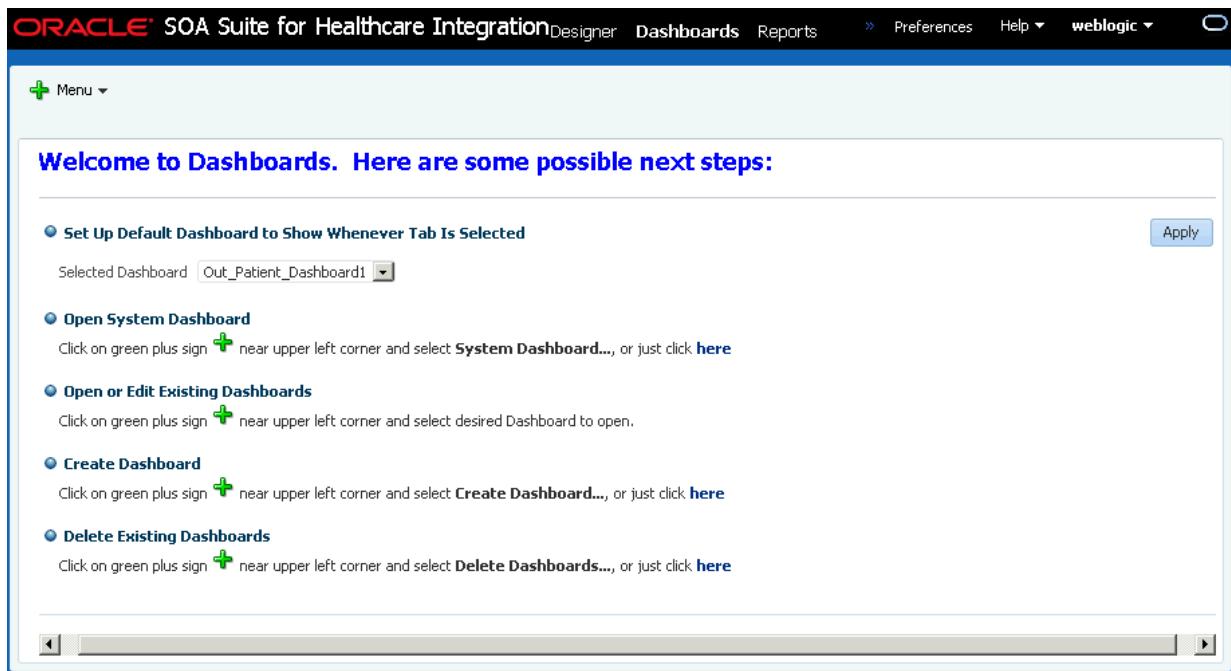
3. Click **Apply**.

   The selected dashboard appears. The next time you log in and select the Dashboards tab, this dashboard automatically appears.

8.3.3 Configuring an Existing Dashboard

After you create a dashboard, you can change the endpoints for which information is displayed, how the information appears on the dashboard, and the type of information that appears.

To configure an existing dashboard

1. If the dashboard you want to modify is not already displayed, click the plus button in the upper right of the main Dashboards page and select the dashboard to configure.

2. When the dashboard displays, click **Show Customizer**.

The customizer appears on the right side of the dashboard.

Figure 8-8   Endpoint Dashboard Customizer
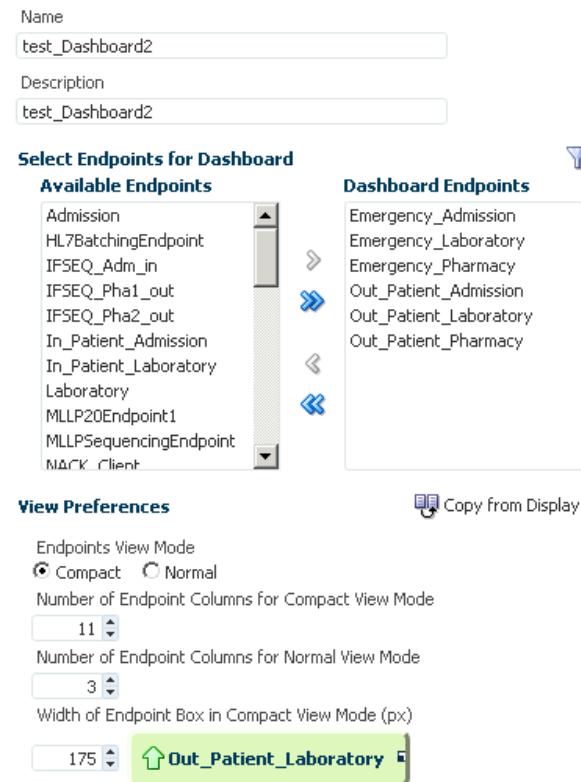
3. Do any of the following:

- Change the name or description of the dashboard.

- Add an endpoint by selecting it in the Available Endpoint section and clicking the right-facing arrow.

- Remove an endpoint by selecting it in the Dashboard Endpoints section and clicking the left-facing arrow.

- Reorder the endpoints by selecting an endpoint to move and clicking the up and down arrows to the right of the customizer.

4. When you are done with your changes, click **Apply** and then click **Hide** to close the customizer.

> **Tip:**
>
> If you do not see your changes in the dashboard after you close the customizer, click the **Refresh** button.

5. To undo the changes you made, click **Revert**.

6. To limit the types of messages displayed on the dashboard, click the down arrow next to the **Show** field, and select one of the following options:

- **All**: Shows all types of messages.

- **Running**: Shows all running endpoints. Idle, disabled, and errored endpoints are not shown.

- **Errors**: Shows only error messages.

- **Disabled**: Shows only disabled endpoints.

Figure 8-9   Show Field on the Dashboards Page
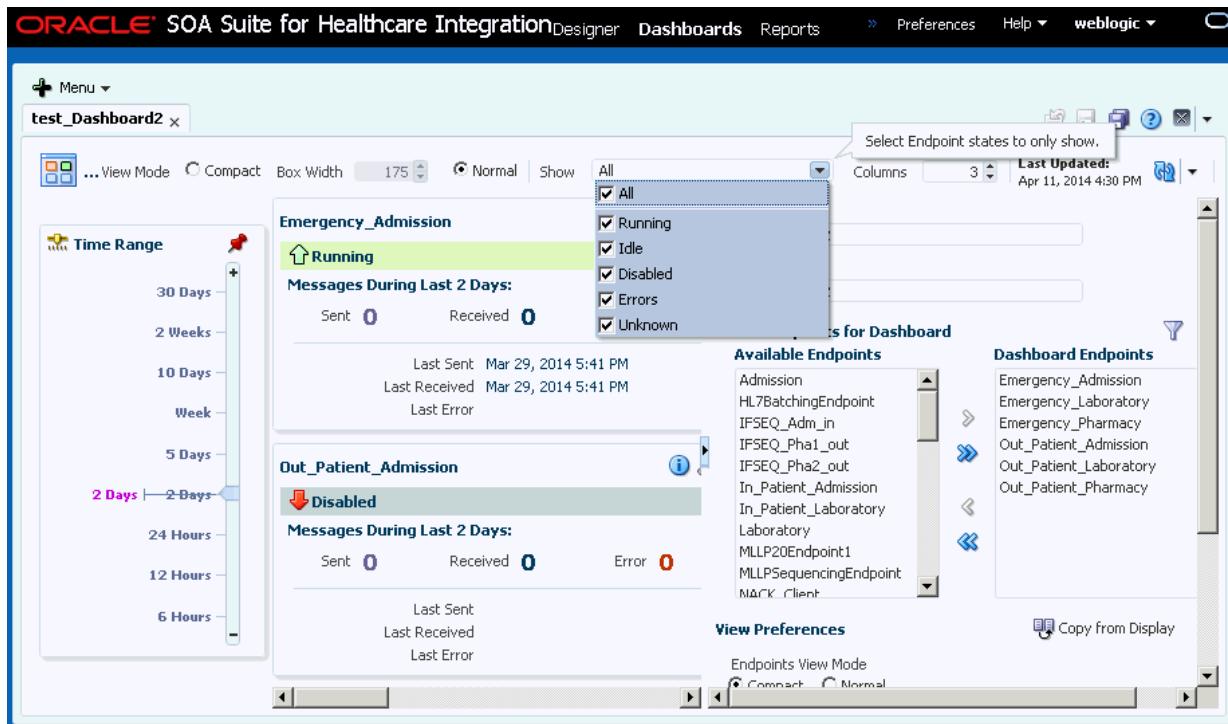
7. To change the layout of the endpoints on the dashboard, change the number of columns displayed in the **Columns** field.

Note:

If you change the number of columns to display for a dashboard, the new value is not persisted when you logout and log back in again.

8.3.4 Refreshing a Dashboard and Setting the Auto-Refresh Rate

You can manually refresh the dashboard at any time, but you can also specify that the dashboard be automatically updated at set intervals.

To refresh a dashboard and set the auto-refresh rate

1. If the dashboard you want to modify is not already displayed, click the plus button in the upper right of the main Dashboards page and select the dashboard to configure.

2. To refresh the dashboard, click the **Refresh** button in the upper right.

3. To enable the auto-refresh option, do the following:

    a. Click the down arrow to the right of the **Refresh** button.

    b. Specify the time interval in seconds to wait between automatically refreshing the dashboard, and then select **Auto-Refresh**.

    c. To disable automatic refreshing, clear the **Auto-Refresh** check box.

8.3.5 Deleting a Dashboard

You can delete any of the dashboards you create. Use caution when deleting dashboards. This action is irreversible.

To delete a dashboard

1. From the main Dashboards page, click the plus button in the upper right and select **Delete Dashboards**.

The Delete Dashboard dialog appears.

Figure 8-10   Delete Dashboards Dialog
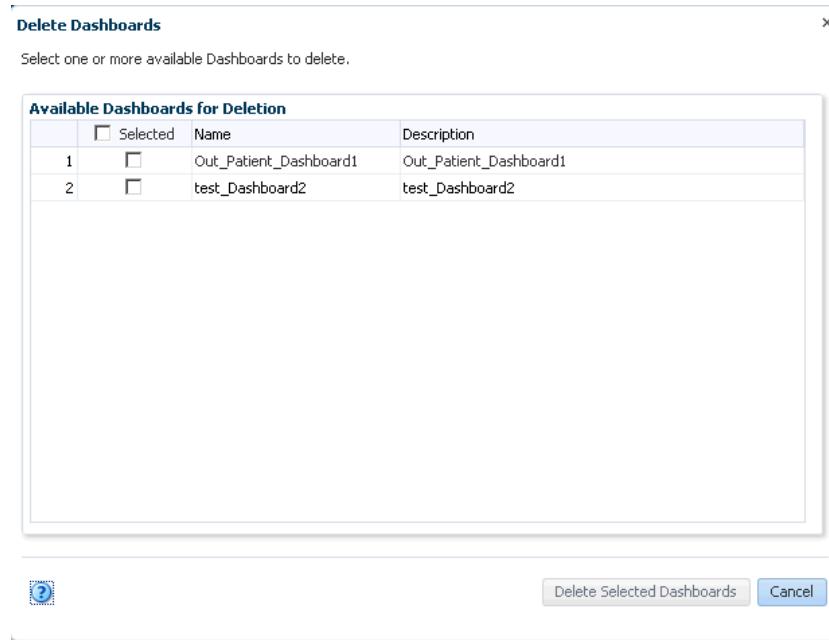
2. In the dashboards list, select the check box next to the dashboards you want to delete.

3. Click **Delete Selected Dashboards**.

The selected dashboards are deleted immediately.

8.4 Viewing Information in Dashboards

Dashboards display a variety of information. You start out viewing a summary of all the endpoints in the dashboard. This view gives you a quick picture of the status of each endpoint, the number of messages they are processing, and whether there were any errors. It also shows you when the last messages were sent or received, and when

the last error occurred. This information can provide clues about which endpoints might require more detailed monitoring.

You can also view more detailed information for each endpoint, including the rate at which messages are being processed, the average message size, the number of messages sent or received, the number of errors, and any error messages. You can also access the Endpoints page directly from the dashboard so you can modify the configuration of an endpoint if required.

8.4.1 Viewing Endpoint Summary Information in a Dashboard

A dashboard has multiple views and you can navigate from each view to get a higher level of detail about a specific endpoint, error, or message.

To view endpoint summary information

1. On the main Dashboards page, click the plus button and then select the dashboard to view. For each endpoint, you can view the following information:

   Figure 8-11   Endpoint Summary on a Dashboard

   - **Status**: The current status of the endpoint, such as Running, Idle, Disabled, or Errors.

   - **Messages Sent**: The number of messages sent by the endpoint in the specified time period.

   - **Messages Received**: The number of messages received by the endpoint in the specified time period.

   - **Errors**: The number of messages with errors for the endpoint in the given time period.

   - **Last Sent**: The date and time the last message was sent from the endpoint.

   - **Last Received**: The date and time the last message was received from the endpoint.

   - **Last Error**: The date and time of the last error for the endpoint.

2. To change the time period for which the dashboard displays information, slide the pointer up or down on the time scale on the left of the page.

Figure 8-12   Sliding Timescale for Dashboard Information

3. To view the properties for a specific endpoint, click or hover over the information button for that endpoint. (Clicking opens a pop-up window with the information; hovering only shows the information while the cursor is over the information button.)

    The Endpoint Properties dialog appears, where you can view the endpoints protocol, connection, polling, timeout, and sequencing properties.

Figure 8-13   Endpoint Properties Dialog, Accessed from a Dashboard

Endpoint Properties - TestFile

Key	Value
Transport Protocol	FILE
Polling interval	5
Folder name	/tmp/in
Identify TP by Delivery Channel	true
Sequencing	false

If the endpoint defines both inbound and outbound processing, the properties dialog has two tabs, one that displays properties for inbound and one that displays properties for outbound. If the statuses of the inbound and outbound are different, additional information about the status appears on the properties dialog.

Figure 8-14   Endpoint Properties Dialog for a Bidirectional Endpoint

For more information about endpoint properties, see Working with Endpoints..

8.4.2 Viewing Detailed Endpoint Information in a Dashboard

From any endpoint summary view, you can navigate into more information for a specific endpoint, such as the message processing rate, the message volume broken down by message type (sent, received, or error), and the average message size. Additional information is available for errors.

To view detailed endpoint information

1. Display the dashboard you want to view on the Dashboards page.

2. For a specific endpoint, click **Show More Endpoint Details** (the button on the far right in the endpoint box).

   The Endpoint Details page appears.

Figure 8-15   Endpoint Details Page on a Dashboard
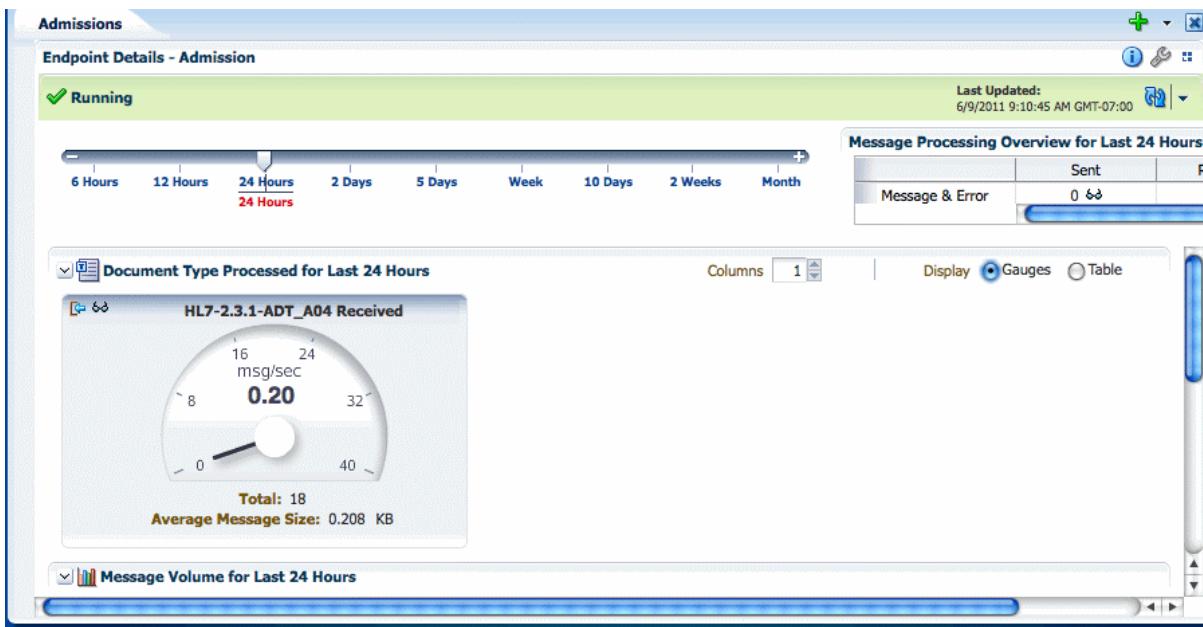
3. To change the time period for the displayed data, move the slider to the left or right on the timescale at the top of the page.

4. You can view the volume metrics for the endpoint as a gauge (Figure 8-15) or in tabular format. Select either **Gauge** or **Table** next to **Display**.

   Both the gauge and the table show the following information:

   - The document type.

   - The number of messages received per second.

   - The total number of message processed in the specified time period.

   - The average size of each message.

5. Scroll down to view a graph showing the volume of sent, received, and error messages.

   Figure 8-16   Graphical Representation of Message Volume

6. Scroll below the graph to view information about sequenced message processing.

Figure 8-17   Sequenced Messages Section of the Dashboard
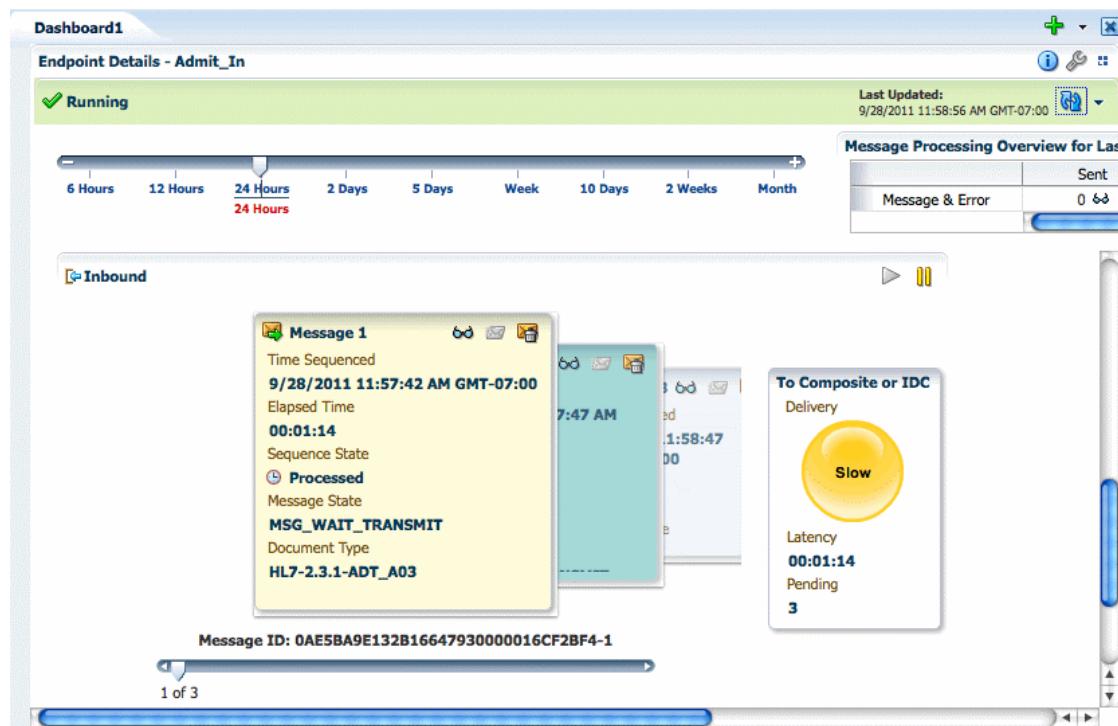
> **Tip:**
>
> If you do not have the sequencing targets configured, you cannot view information about sequencing on the dashboard. For information on configuring sequencing, see Enabling Sequencing for an MLLP Endpoint.

7. Scroll to the bottom of the page to view error messages for the displayed endpoint.

8. To view a report of the messages processed, click the eyeglass button either in the gauge or on the table. You can also view a report by clicking the eyeglass button in the **Message Processing Overview for Last 24 Hours** box. Both actions open a new tab showing the report.

Figure 8-18   Endpoint Report Accessed from a Dashboard

For more information about reports and the information displayed, see Working with Reports .

8.4.3 Configuring an Endpoint from a Dashboard

The dashboard lets you access the endpoint configuration pages directly so you can view and edit the endpoint properties.

To configure an endpoint from a dashboard

1. Display the dashboard you want to view on the Dashboards page.

2. In the box for the endpoint you want to configure, click **Configure This Endpoint** (the wrench button).

    The Endpoint page appears.

3. Modify any of the endpoint values, as described in "Working with Endpoints."

Note:

This takes you out of the dashboard pages. To return to the dashboard you were viewing, click the Dashboards tab.

8.5 Working with Sequenced Messages

The Sequenced Messages section of an endpoint dashboard provides information about the health of the sequenced message queue. In this section, you can view information about the messages that are pending in the sequenced message queue. If there is no information in this section, then sequenced messages are processing

successfully. If messages are backed up, you can view the necessary information and take steps to unblock the queue.

The information you can view for the queue includes the number of messages in the queue, the time a message was sequenced, the time elapsed since sequencing was initiated for the message, the sequence state, the message processing state, and the document type. There is also an indicator of the overall delivery status for the queue. Possible statuses are:

- **Paused**: Indicated by a red traffic light, this means that sequencing has been manually paused or blocked when the message at the front of the queue has encountered a transport error.

- **Slow**: Indicated by a yellow traffic light, this means that the elapsed time for the message at the front of the queue has exceeded the `Slow Threshold for Sequencing` property in the UI Settings.

- **Running**: Indicated by a green traffic light, this indicates that sequenced messages are being processed normally.

Messages can get backed up in the queue due to sequencing errors, message errors, or message processing delays. The Sequenced Messages section gives you the information you must correct these problems. You can view information about messages in the queue to determine the cause of the stoppage. After you find the source, you can reprocess errored messages or discard messages from the queue.

To work with sequenced messages

1. Display the dashboard you want to view on the Dashboards page.

2. For a specific endpoint, click **Click to show Maximized Endpoint Details** (the button on the far right in the endpoint box).

    The Maximized Endpoint Details page appears.

3. Scroll down to the Sequenced Messages section.

    If there are messages waiting in the sequencing queue, you can view them here.

Figure 8-19   Sequenced Messages Section of the Endpoint Dashboard
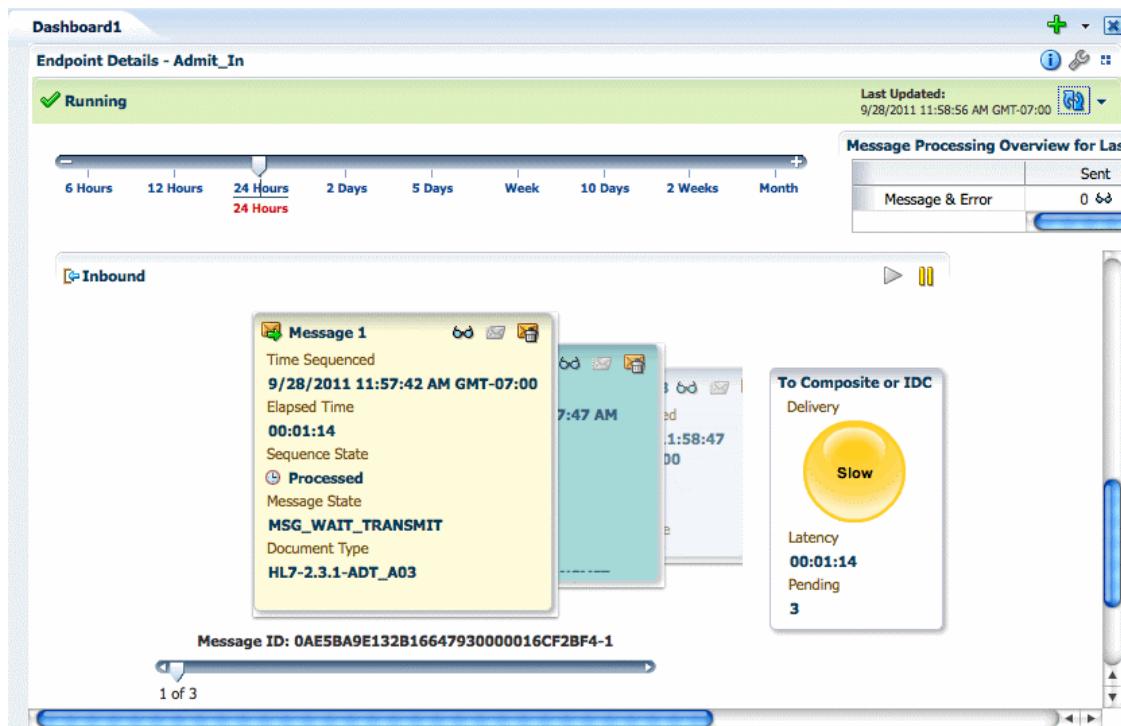
4. The waiting messages appear as a carousel of note boxes. Click through the boxes or use the slider underneath the messages to view information about each message.

5. To view a report of any of the messages, click the eyeglass button in the upper right of that message's box.

   The report page appears for the selected message.

6. When sequencing is blocked by a message, you can do any of the following:

   • View the message report, as shown in Figure 8-19, and use the Purge or Resubmit functions of the reports page to handle the message.

   • In the Sequenced Message section, reprocess an errored message by clicking the **Reprocess an Errored Message** button in the upper right of the message box.

   • In the Sequenced Messages section, discard the blocked message by clicking the **Discard Message** button in the upper right of the message box. Discarding a message only removes it from the sequenced message queue, and does not remove it from the instance data shown in the reports.

7. To pause sequenced message processing, click the **Pause** button in the upper right of the Sequenced Messages section.

8. To resume sequenced message processing, click the **Resume** button in the upper right of the Sequenced Messages section.

8.6 Viewing Endpoint Error Messages

When a message for a specific endpoint generates an error, you can access information about the error from the Dashboards page.

Note:

When a transport error occurs in an endpoint, the status bar for the endpoint changes to red and the status changes to **Errors**. This does not indicate an error in the business message processing, which is described in the following procedure, though message processing errors can include transport errors as well. Message processing errors are indicated by the number in the **Error** field of the endpoint summary.

Figure 8-20 Endpoint Summary with Errors
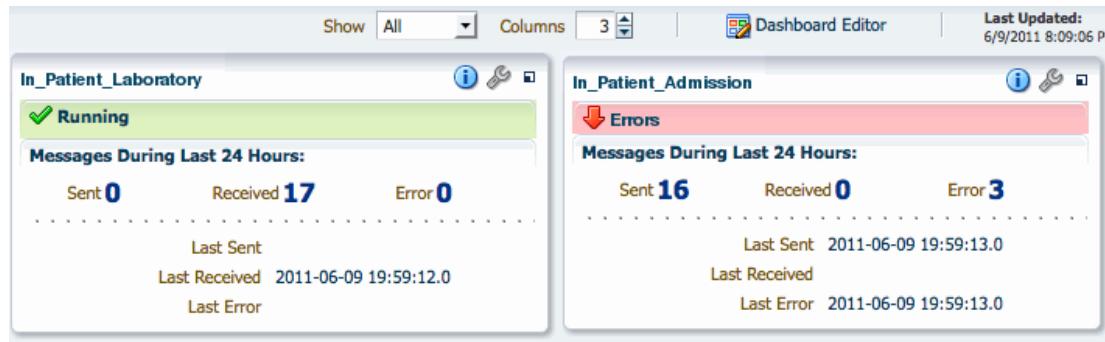
To view endpoint error messages

1. On the endpoint summary dashboard page, click **Show More Endpoint Details** for the endpoint that shows messages with errors.

2. On the endpoint details page, scroll to the bottom of the page to view the list of error messages.

Figure 8-21 Endpoint Error Messages List

3. To view a report of the error, click the eyeglass button to the right of the message ID.

The Reports page appears, as shown in Figure 8-22.

Figure 8-22   Endpoint Error Message Report

4. To view more details about the error, collapse the upper portion of the window by clicking the up arrow beneath the message list or click **Pin Current Message Details Into a New Tab** to open the details in a new tab.

The business message details appear.

Figure 8-23   Business Message Details

For more information about reports and the information displayed, see Working with Reports .

5. In the flow diagram, click the red X button that indicates where the error occurred. The Error Details appear below the diagram.

Figure 8-24   Endpoint Transport Error Details

9
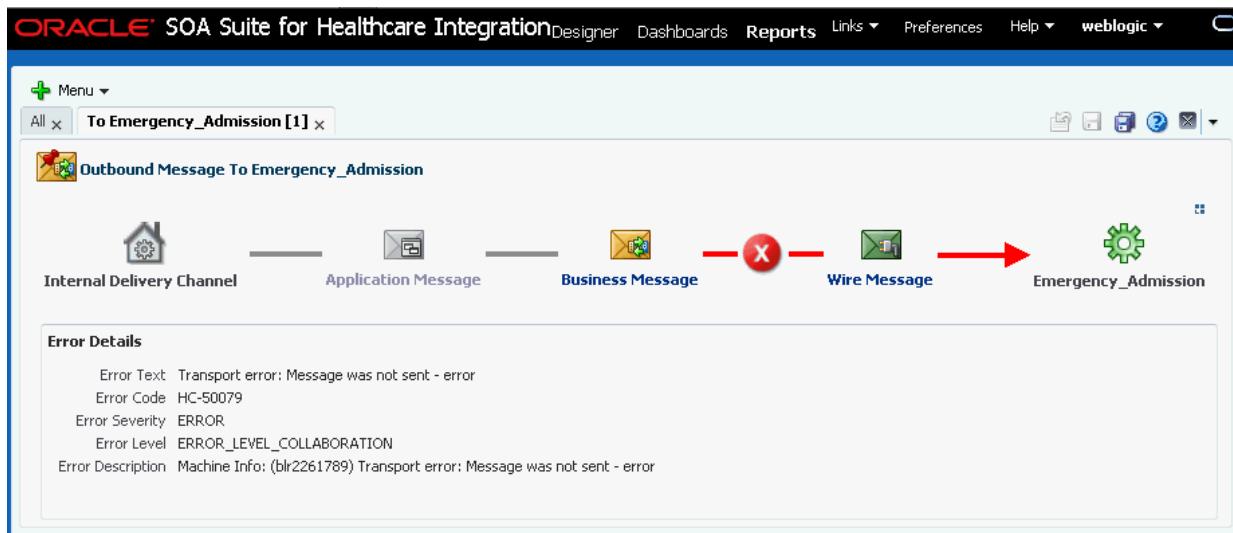
Working with Reports

This chapter describes the reporting features of Oracle SOA Suite for healthcare integration. The configurable reports display the real-time status of message processing through the healthcare integration application based on the criteria you specify.

This chapter includes the following topics:

- Introduction to Reports

- Creating and Configuring Reports

- Viewing Reports and Report Information

- Working with Reports for Unassociated Messages

- Working with Error Messages

- Purging Messages from the Repository

- End-to-End Monitoring of Runtime Data

9.1 Introduction to Reports

The Reports page of the Oracle SOA Suite for healthcare integration user interface lets you view the status of the messages being processing through Oracle Healthcare components in real-time. You can view all messages, or you can narrow down the messages displayed by a variety of criteria, including endpoints, date ranges, document information, and payload fields.

Several reports are predefined on the Oracle Healthcare user interface. These are Wire Message report, Unassociated Wire Message report, Business Message report, Application Message report, and Unassociated Application Message report. You can create additional reports based on more specific criteria to narrow down the types of messages displayed.

Each report is divided into four sections:

- A report configuration bar

- The message list

- A graphical depiction of the message flow

- Message details for the selected message

If you select multiple messages from the report, a summary of information for the selected messages appears in place of the flow diagram.

Oracle Healthcare provides a set of predefined reports based on the following report types that you can access by clicking the plus button on the top right corner of the Reports home page:

- General Business Message reports: Lists all the Business Messages that have been exchanged from the beginning or in a give time range. You can customize these reports by using the Message Report Filter Customizer.

- General Application Message reports: Lists all the Application Messages that have been exchanged. You can customize the report by using the Message Report Filter Customizer.

- General Wire Message reports: Lists all the Wire Messages that have been exchanged. You can customize the report by using the Message Report Filter Customizer.

- Unassociated Application Message reports: Lists all the Application Message reports that display Application Messages without any associated Business Messages. You can customize these reports by using the Message Report Filter Customizer.

- Unassociated Wire Message reports: Lists all the Wire Message reports that display Wire Messages without any associated Business Messages. You can customize these reports by using the Message Report Filter Customizer.

See Working with Reports for Unassociated Messages for more information on Unassociated Message reports.

The customized reports are displayed in the Oracle Healthcare console under their respective report types as shown in Figure 9-1.

Figure 9-1   Types of Reports
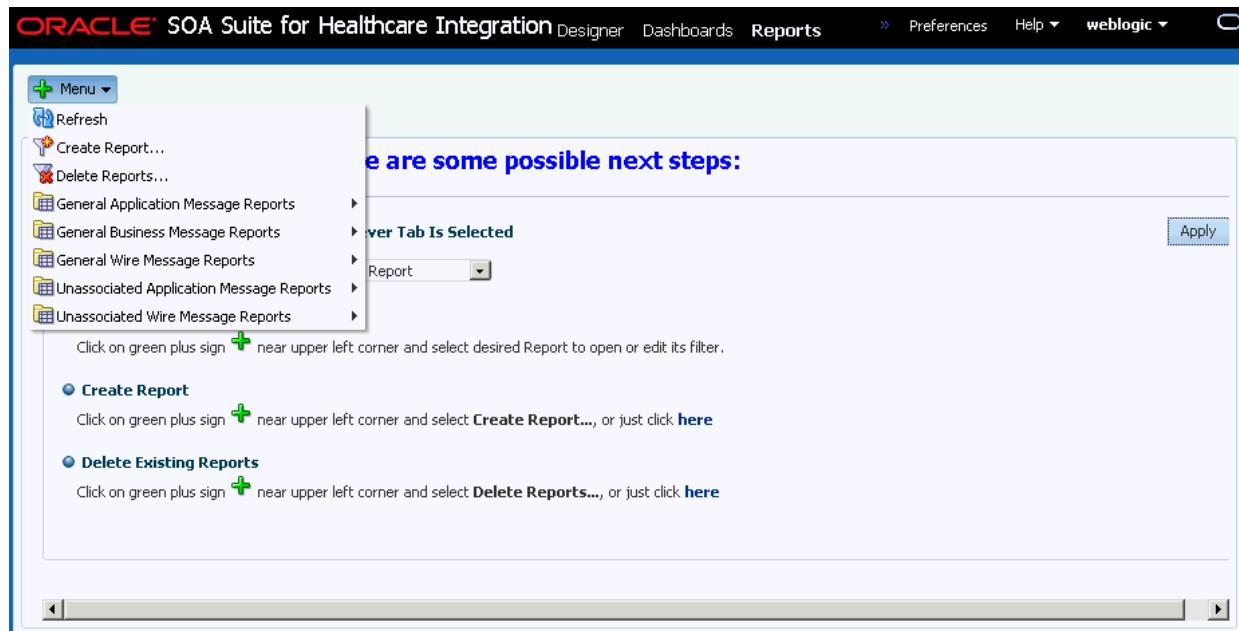
Figure 9-2 shows an example of a General Business Message report as displayed in the Reports tab.

Figure 9-2   Reports Page Showing a Predefined Business Message Report

Figure 9-3 shows an example of a General Wire Message report as displayed in the Reports tab.

Figure 9-3   Reports Page Showing a Predefined Wire Message Report

Figure 9-4 shows an example of a General Application Message report as displayed in the Reports tab.

Figure 9-4   Reports Page Showing a Predefined Application Message Report

In the message list, each message is marked with a button that provides processing information about the message. An envelope button indicates normal processing; a green arrowhead over the envelope indicates the message was resubmitted; and a red square with an "X" over the envelope indicates an error.

9.1.1 About the Message Report Filter Customizers

You use the three Message Report Filter Editors (Wire, Business, and Application) to create reports and to edit existing reports. The report editors allow you to define a wide variety of criteria for the messages that are displayed in the report, including a time range, endpoint, protocol, state, message status, and document definition information. You can also use advanced options, which include correlation fields, SOA composite fields, message IDs, and so on.

Figure 9-5 displays a sample Business Message Report Customizer.

Figure 9-5   Business Message Report Filter Customizer

For most filters, you must select an operator that specifies how to evaluate the value you specify for the filter. You can also specify whether or not the search is case-sensitive. Most filters support the following operators to act against the value you specify: **Any**, **Equals**, **Like**, **Not Equal**, **Among**, or **Not like**. For the **Among** operator, you specify a list of up to 100 values, separated by commas, and messages that match one of the values in the list will be filtered. For the **Like** and **Not Like** operators, you can use SQL wildcard characters in the value. SQL wildcard characters include:

- **%** (percent): Represents zero or more unknown characters

- **_** (underscore): Represents a single character

- **[charlist]**: A list of characters; the unknown value is any character in the list

- **[^charlist]** or **[!charlist]**: A list of characters; the unknown value is *not* a character in the list

For most string-based criteria, you can specify whether or not the search should be case sensitive by selecting or deselecting **Ignore case** next to the field. Use the links next to **Filter Conditions** to move between sections in the filter conditions list.

9.1.2 About Resubmitting Messages

When messaging errors are internal to Oracle SOA Suite for healthcare integration, you can correct the problem and resend the message. For example, if a message is sent to an endpoint that is not configured correctly, correct the error and use the resubmit feature for application messages or wire messages.

Resubmitting an application message for an outbound message clones the message, assigns the new message a status of RESUBMITTED, and attempts to deliver the clone. In addition, the RESUBMIT_COUNT of the original application message is incremented by 1. The RESUBMIT_REF_TO of the cloned message is set to the ID of the original application message as well. If the resubmitted application message is part of the batch message, the state of the associated wire message is set to RESUBMITTED as well. Resubmitting this type of message is helpful when the document configuration is not as per requirement and the message must be restructured with updated settings.

Resubmitting an application message for an inbound message sets the status of the message to RESUBMITTED and attempts to deliver the message again to the back-end application. Resubmitting this type of message is useful when the back-end application is down and the delivery must be retried.

Resubmitting a wire message for an outbound message sets the status of the message to RESUBMITTED and attempts to redeliver only the previously processed message. There is no repackaging or other message transformation. This is helpful when the problem was with the delivery endpoint (for example, the receiver's server is down and unable to receive the message).

Resubmitting a wire message for an inbound message clones the wire message, assigns the new message a status of RESUBMITTED, and attempts to deliver the message. In addition, the RESUBMIT_COUNT of the original wire message is incremented by 1. The RESUBMIT_REF_TO of the cloned message is set to the ID of the original wire message as well. The message statuses of the business and application messages are also set to RESUBMITTED. The functional acknowledgment is not returned to the remote endpoint even though the endpoint expects it. This is useful when the document settings are not correct and the message must be translated and validated again.

9.1.3 Important Note About Clustered Environments

In a clustered environment, if system time stamps are not synchronized for all nodes in the cluster, you might see message time stamps that look incorrect, but are not. For example, given an unsynchronized, multi-node cluster, if an outbound message is received on one node, but the reply is sent from another node, it is possible for a report to show message receipt at 4 a.m., but an acknowledgment sent at 3:55 a.m.

9.2 Creating and Configuring Reports

Oracle SOA Suite for healthcare integration provides several predefined reports for you to get started with. These reports are filtered only by a time range that you can configure under Designer > Administration > Settings > UI > Reports > Time Range.

See Configuring the User Interface Settings for more details. You can create additional reports that further filter messages by criteria such as endpoint, message ID, document properties, composite properties, transport protocol properties, and key payload fields. You can also use the predefined reports as a basis, and either modify them or save them as new reports.

9.2.1 Creating Business Message Reports

Although you have a predefined Business Message report, you can create additional reports to meet your specific requirements.

To create a Business Message report

1. On the Oracle SOA Suite for healthcare integration user interface, click the **Reports** tab.

   The Reports Welcome page appears when there are no opened reports.

Figure 9-6   Reports Welcome Page
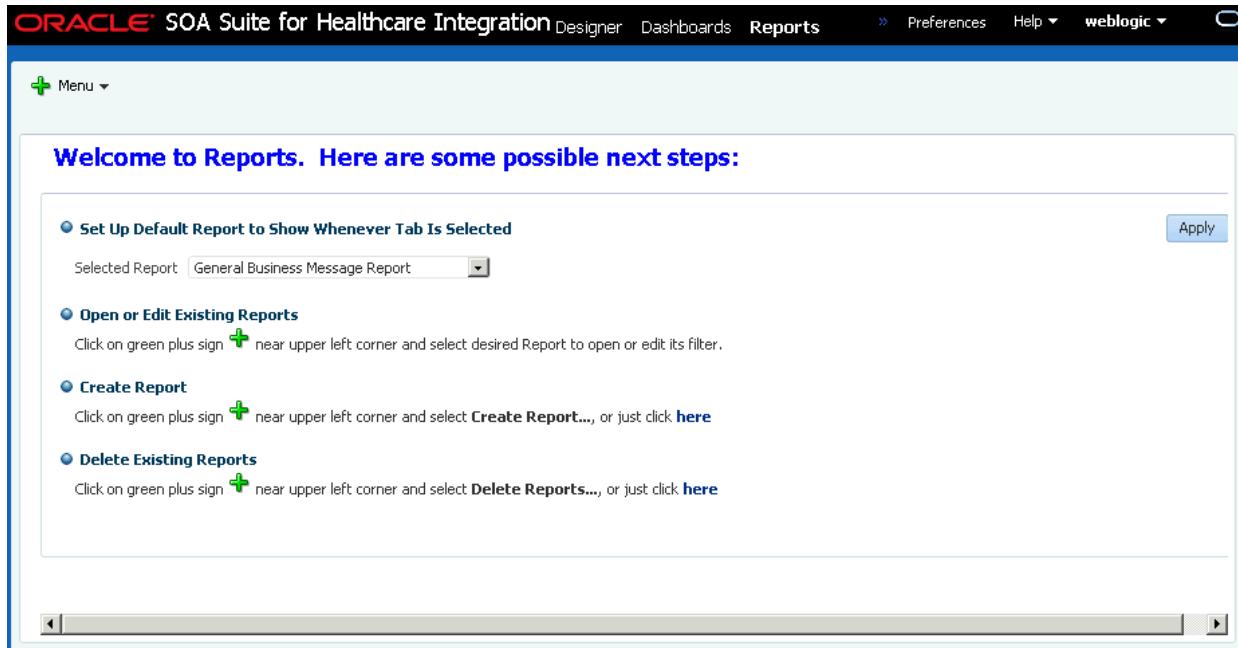
2. Click the plus button in the upper right, and select **Create Report.**

3. On the Create Message Report dialog, enter a name for the report, select **General Business Message Reports** from the **Report Type** list, and click **Create** as shown in Figure 9-7.

   Figure 9-7   Create Report
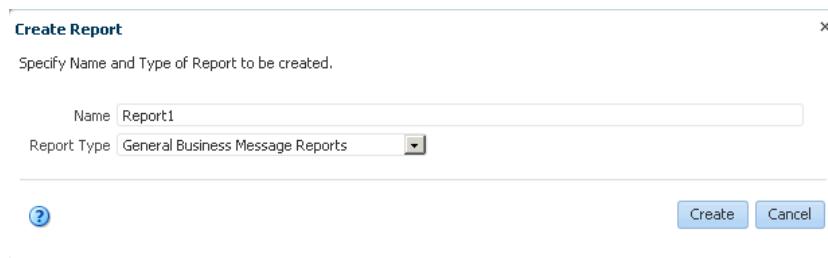
   The report appears with the Message Report Filter Customizer displayed.

4. On the Message Report Filter Customizer, enter any of the filter criteria described in Table 9-1.

   Note:

   When a Wire/Business/Application Message Report is first created (or cloned by doing a Save As New), no search is automatically done when the report is first opened. You must explicitly click the **Search** button.

Use the links next to **Filter Conditions** to quickly move between the different sections in the filter conditions list.

> **Note:**
>
> Multiple conditional operators are supported for most string criteria. For more information about using **Any**, **Equals**, **Like**, **Among**, **Not Equal**, or **Not Like**, see About the Message Report Filter Customizers.

Table 9-1   Business Message Report Customizer Options

Filter	Description
Name	The name of the report.
Description	A brief description of the report.
Match	An indicator of whether the search criteria are joined by an AND or OR operator. Select **Match all of the following conditions** to use AND; select **Match any of the following condition** to use OR.
Message ID (Business)	The business message ID of the message to display. Select the operator and then enter an appropriate value for a message ID.
Any	Does not narrow the report by any date range, and displays messages from all dates and times.
Last	Narrows the report down to a range of most recent dates or times. For example, you can select the past 12 hours, past five days, or past two weeks. Slide the pointer on the sliding scale to select a time range. You can modify the Enabled Time Slider Periods under **UI Settings**.
Range	Narrows the report down to the range of dates you specify. In the **From** or **To** field, or in both fields, provide a date and time in the format shown (MM/DD/YYYY HH:MM:SS AM/PM) or click the **Select Date and Time** button to select a date and specify the time.
	To search for all messages after a specific date, only enter a date for the **From** field. To search for all messages earlier than a specific date, only enter a date in the **To** field. To search for all messages within a range of dates, enter dates in both the **From** and **To** fields. To exclude a specific time range, enter the first date in the range to exclude in the To field and enter the last date in the range to exclude in the From field. You must select the **Match any of the following conditions** indicator to use this excluded range. The search returns all messages received.
	For both the from and to dates, select **Inclusive** to include the specified dates in the search, or clear **Exclusive** to exclude those dates from the search.
Direction	The direction of the messages to display. Select from **Any**, **Equals**, or **Not Equals**, and then click in the field on the right to select a direction (inbound or outbound) from the list

Table 9-1   (Cont.) Business Message Report Customizer Options

Filter	Description
Endpoint	The endpoint from where the messages are received or to which the messages are delivered. Select the operator and then enter the name of a defined endpoint.
State	The state of the messages to display. Select from **Any**, **Equals**, or **Not Equals**, and then click in the field on the right to select one or more message states from the list.
Document	The following three fields specify document properties. These three fields control what is shown in the Document Type column in the reports table. For any of these fields, select **Ignore case** if you do not want the search to be case sensitive.
• Protocol	The document protocol of the messages to display. Select the operator and then enter the document protocol. For HL7 messages, enter **HL7**.
• Version	The document protocol version of the messages to display. Select the operator and then enter the version. For example, for HL7 2.6 messages, enter **2.6**.
• Type	The document protocol type of the messages to display. Select the operator and then enter the version. For example, for HL7 ADT_A04 messages, enter **ADT_A04**.
Payload Key Fields	Name and value pairs that narrow the search down by the contents of a field in a message. For each field name you enter, enter a corresponding value for the field. For any of the names or values, select the operator.
	These fields can only be used if they are defined on the XPath page for the document definition. For **Name** fields, enter the name specified on the XPath page, which represents a field in the message. For **Value** fields, enter the expected value of the corresponding field.
	When you create your document definitions, you can specify up to three fields in the message as payload key fields. These fields are defined by a unique name and an XPath expression that locates the field in the message. For example, you might name a field **LastName** and specify the XPath expression for the last name field in the PID segment of an HL7 A03 message. You can similarly define correlation fields for a document definition.
	If you have defined either correlation fields or payload key fields for a document definition, then you can use these fields as filters for business message reports. In the filter editor, these filters are defined by name and value pairs. You use the unique name you gave the field when you created the document definition, and then specify what the value should be for that field in the messages you want to include on the report.
Document	
• Document Definition	The document definition used by the messages to display. Select the operator and then enter the name of a defined document definition.

Table 9-1   (Cont.) Business Message Report Customizer Options

Filter	Description
• Interchange Control Number	A unique identifier for the interchange. This number is used to track messages, to find duplicate or missing transmissions, or to find messages that are out of sequence.
• Group Control Number	A unique identifier for the group. This number is used for auditing, to find duplicate or missing groups, or to find groups that are out of sequence.This number is referenced by the functional acknowledgment.
• Transaction Set Control Number	A unique identifier for the transaction set. The functional acknowledgment references this number for transaction set acknowledgment.
Composite	
• Transport Protocol Name	The name of the transport protocol used by the Oracle SOA composite application.
• Transport Protocol Version	The version of the transport protocol used by the Oracle SOA composite application
• ECID	The execution context ID. The ECID enables end-to-end message tracking.
• Composite Instance ID	The Oracle SOA composite instance ID.
• Composite Name	The name of the Oracle SOA composite application in Oracle JDeveloper.
• Service Name	The name of the healthcare integration service binding component in the Oracle SOA composite application.
• Reference Name	The name of the healthcare integration reference binding component in the Oracle SOA composite application.
• Domain Name	The name of the WebLogic domain on which the Oracle SOA composite application is deployed.
• Composite Version	The version of the Oracle SOA composite application in Oracle JDeveloper.
Correlation	
• Correlation From XPath Name	The name of the correlation field for initiating the correlation This and the remaining correlation fields can only be used if they are defined on the Correlation page for the document definition. For usage information, see **Payload Key Fields** in Table 9-1.
• Correlation From XPath Value	The value of the above field.
• Correlation To XPath Name	The name of the second field for the correlation.
• Correlation To XPath Value	The value of the above field.

Table 9-1   (Cont.) Business Message Report Customizer Options

Filter	Description
Error	
• Error Code	The error code for error message to display on the report.
• Error Text	The text of an error message to display on the report.
Message	
• Application Message ID	The unique identifier for the application message to display in the report.
• Protocol Message ID	The unique protocol message identifier for messages displayed in the report.
• Native Message Size	The size of the original message prior to being translated. Select from **Any**, **Equals**, **Not Equal**, **Less than**, **Less than or equals**, **Greater than**, or **Greater than or equals**, and provide a numerical value.
• Translated Message Size	The size of the message after it is translated. Select from **Any**, **Equals**, **Not Equal**, **Less than**, **Less than or equals**, **Greater than**, or **Greater than or equals**, and provide a numerical value.

5. To revert any unsaved changes you made to the report filters, click **Revert**.

6. When you are done specifying report filters, do any of the following:

   • To test your filter criteria, click **Search**, and then click the **Show/Hide Filter** toggle button.

   • To save the changes you made to the filter criteria for the report, click **Save**.

   > **Note:**
   >
   > You cannot save changes made to the standard reports (such as General Business Message Report). You can only click the **Save As New** button that is enabled only when the name of the report has been changed from its standard form.

Figure 9-8 displays a custom Business Message Report, which lists all the Business Messages exchanged by Emergency_Admission and Emergency_Laboratory endpoints in the last 30 days.

Figure 9-8   A Custom Business Message Report

You can sort the messages based on any of the report columns by clicking the column headers.

9.2.2 Creating Wire Message Reports

Apart from the predefined Wire Message reports, you can create customized Wire Message reports that covers Wire Messages coming from and going to an external endpoint. Most Wire Messages have associated Business Messages, which is reflected in the Message Flow diagram in the report.

To create a Wire Message report

1. Perform Steps 1 and 2 from Creating Business Message Reports.

2. On the Create Message Report dialog, enter a name for the report, select **General Wire Message Reports** from the **Report Type** list, and click **Create**.

   The report appears with the Message Report Filter Customizer displayed as shown in Figure 9-9.

Figure 9-9   Wire Message Report Filter Customizer
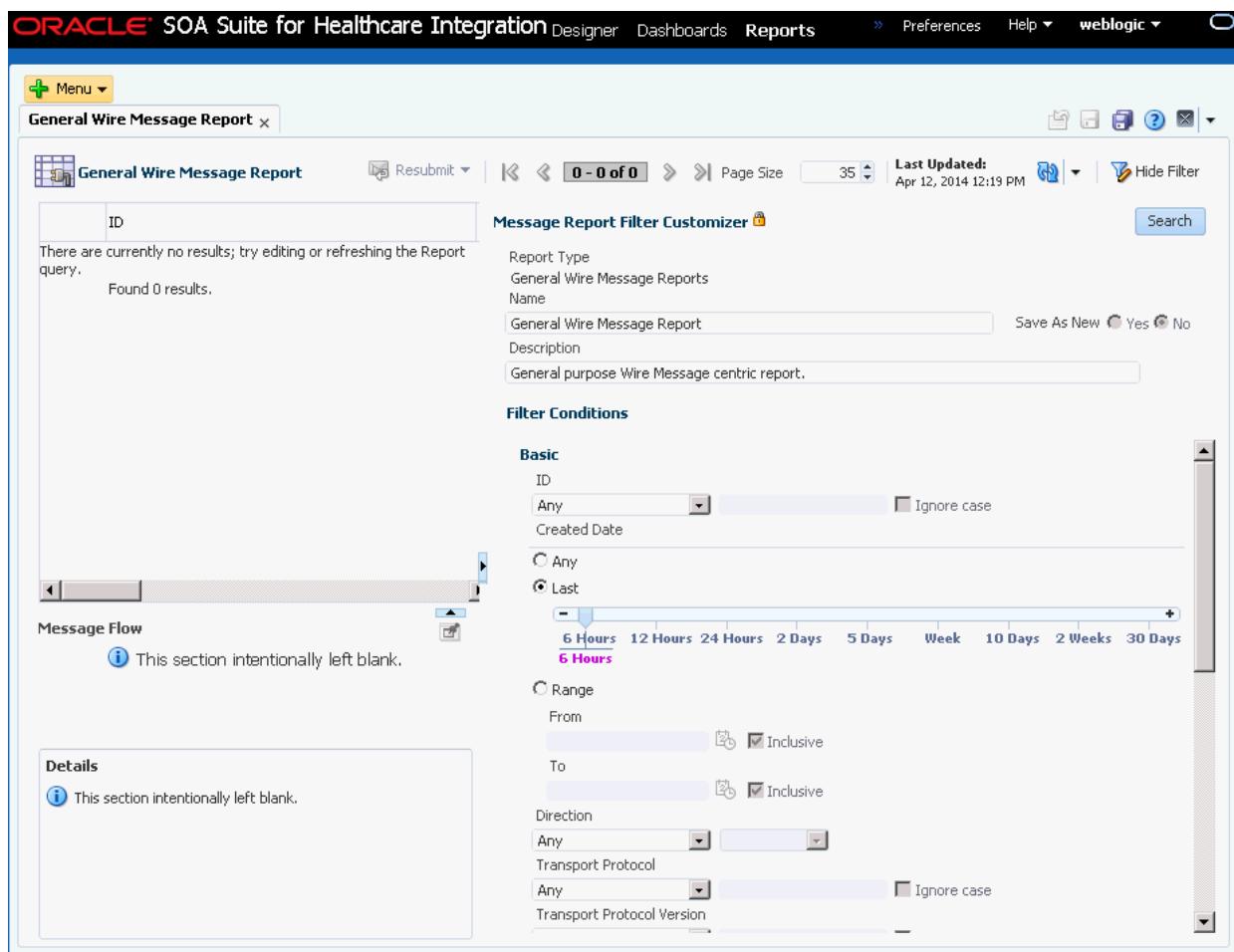
3. On the Message Report Filter Customizer, enter any of the filter criteria described in Table 9-2.

Note:

Multiple conditional operators are supported for most string criteria. For more information about using **Any**, **Equals**, **Like**, **Among**, **Not Equal**, or **Not Like**, see About the Message Report Filter Customizers.

Table 9-2   Wire Message Report Filter Customizer Options

Filter	Description
Name	The name of the report.
Description	A brief description of the report.
ID	The Wire Message ID of the message to display. Select the operator and then enter all or part of a message ID.
Any	Does not narrow the report by any date range, and displays messages from all dates and times.

Table 9-2   (Cont.) Wire Message Report Filter Customizer Options

Filter	Description
Last	Narrows the report down to a range of most recent dates or times. For example, you can select the past 12 hours, past five days, or past two weeks. Slide the pointer on the sliding scale to select a time range.
Range	Narrows the report down to the range of dates you specify. In the **From** or **To** field, or in both fields, provide a date and time in the format shown (MM/DD/YYYY HH:MM:SS AM/PM) or click the **Select Date and Time** button to select a date and specify the time.
	To search for all messages after a specific date, only enter a date for the **From** field. To search for all messages earlier than a specific date, only enter a date in the **To** field. To search for all messages within a range of dates, enter dates in both the **From** and **To** fields. To exclude a specific time range, enter the first date in the range to exclude in the To field and enter the last date in the range to exclude in the From field. You must select the **Match any of the following conditions** indicator to use this excluded range. The search returns all messages received.
	For both the from and to dates, select **Inclusive** to include the specified dates in the search, or clear **Exclusive** to exclude those dates from the search.
Direction	The direction of the messages to display. Select from **Any**, **Equals**, or **Not Equal**, and then click in the field on the right to select a direction (inbound or outbound) from the list
Transport Protocol	The transport protocol of the messages to display. Select from **Any**, **Equals**, **Like**, **Not equal**, or **Not like**, and then enter the name of a defined transport protocol. Select **Ignore case** if you do not want the search to be case sensitive.
Transport Protocol Version	The transport protocol version of the messages to display. Select from **Any**, **Equals**, **Like**, **Not equal**, or **Not like**, and then enter the name of a defined transport protocol version. Select **Ignore case** if you do not want the search to be case sensitive.
URL	The TCP endpoint URL of the messages to display. Select from **Any**, **Equals**, **Like**, **Not equal**, or **Not like**, and then enter a defined URL value. Select **Ignore case** if you do not want the search to be case sensitive.
State	The state of the messages to display. Select from **Any**, **Equals**, **Like**, **Not equal**, or **Not like**, and then click in the field on the right to select one or more message states from the list.
Message Size (bytes)	The size of the messages (in bytes) to display. Select from **Any**, **Equals**, **Not equal**, **Less than**, **Less than or equals**, or **Greater than**, or **Greater than or equals**, and then enter a defined message size.

Table 9-2   (Cont.) Wire Message Report Filter Customizer Options

Filter	Description
Resubmit Count	The number times a message (to be displayed) has been resubmitted. Select from **Any**, **Equals**, **Not equal**, **Less than**, **Less than or equals**, or **Greater than**, or **Greater than or equals**, and then enter a defined count.
Last Resubmitted Date	The date when a message (to be displayed) was last resubmitted. Select from the **Any**, **Last**, or **Range** options.

4. To revert any unsaved changes you made to the report filters, click **Revert**.

5. When you are done specifying report filters, do any of the following:

   • To test your filter criteria, click **Search**, and then click the **Show/Hide Filter** toggle button.

   • To save the changes you made to the filter criteria for the report, click **Save As New**.

   Figure 9-10 displays a custom Wire Message Report, which lists all the inbound Wire Messages in the last 30 days.

Figure 9-10   A Custom Wire Message Report

You can sort the messages based on any of the report columns by clicking the column headers.

9.2.3 Creating Application Message Reports

Apart from the predefined Application Message reports, you can create customized Application Message reports that covers Application Message going to or coming from the Fabric/Composite layer. Most Application Messages coming into Oracle Healthcare runtime result in corresponding Business Messages, which is reflected in the Message Flow diagram in the report.

To create a Wire Message report

1. Perform Steps 1 and 2 from Creating Business Message Reports.

2. On the Create Message Report dialog, enter a name for the report, select **General Application Message Reports** from the **Report Type** list, and click **Create**.

   The report appears with the Message Report Filter Customizer displayed as shown in Figure 9-11.

Figure 9-11   Application Message Report Filter Customizer
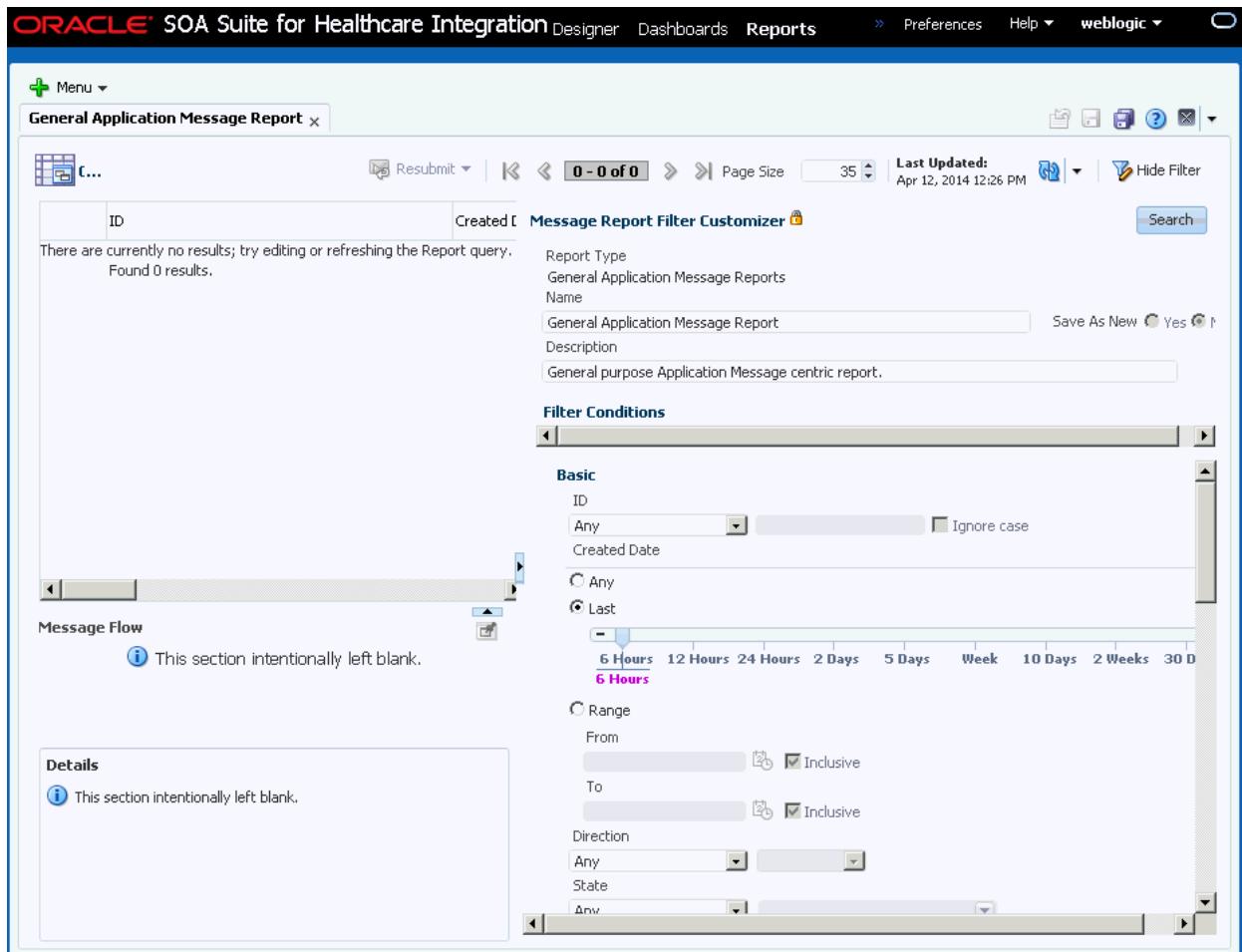
3. On the Message Report Filter Customizer, enter any of the filter criteria described in Table 9-3.

> **Note:**
>
> Multiple conditional operators are supported for most string criteria. For more information about using **Any**, **Equals**, **Like**, **Among**, **Not Equal**, or **Not Like**, see About the Message Report Filter Customizers.

Table 9-3   Application Message Report Filter Customizer Options

Filter	Description
Name	The name of the report.
Description	A brief description of the report.
ID	The Application Message ID of the message to display. Select the operator and then enter all or part of a message ID.
Any	Does not narrow the report by any date range, and displays messages from all dates and times.
Last	Narrows the report down to a range of most recent dates or times. For example, you can select the past 12 hours, past five days, or past two weeks. Slide the pointer on the sliding scale to select a time range.
Range	Narrows the report down to the range of dates you specify. In the **From** or **To** field, or in both fields, provide a date and time in the format shown (MM/DD/YYYY HH:MM:SS AM/PM) or click the **Select Date and Time** button to select a date and specify the time. To search for all messages after a specific date, only enter a date for the **From** field. To search for all messages earlier than a specific date, only enter a date in the **To** field. To search for all messages within a range of dates, enter dates in both the **From** and **To** fields. To exclude a specific time range, enter the first date in the range to exclude in the To field and enter the last date in the range to exclude in the From field. You must select the **Match any of the following conditions** indicator to use this excluded range. The search returns all messages received. For both the from and to dates, select **Inclusive** to include the specified dates in the search, or clear **Exclusive** to exclude those dates from the search.
Direction	The direction of the messages to display. Select from **Any**, **Equals**, or **Not Equal**, and then click in the field on the right to select a direction (inbound or outbound) from the list
State	The state of the messages to display. Select from **Any**, **Equals**, **Like**, **Not equal**, or **Not like**, and then click in the field on the right to select one or more message states from the list.
Message Size (bytes)	The size of the messages (in bytes) to display. Select from **Any**, **Equals**, **Not equal**, **Less than**, **Less than or equals**, or **Greater than**, or **Greater than or equals**, and then enter a defined message size.
Document	

Table 9-3   (Cont.) Application Message Report Filter Customizer Options

Filter	Description
• Protocol	The document protocol of the messages to display. Select the operator and then enter the document protocol. For HL7 messages, enter **HL7**.
• Version	The document protocol version of the messages to display. Select the operator and then enter the version. For example, for HL7 2.6 messages, enter **2.6**.
• Type	The document protocol type of the messages to display. Select the operator and then enter the version. For example, for HL7 ADT_A04 messages, enter **ADT_A04**.
• Document Definition	The document definition used by the messages to display. Select the operator and then enter the name of a defined document definition.
Composite	
• ECID	The execution context ID. The ECID enables end-to-end message tracking.
• Domain Name	The name of the WebLogic domain on which the Oracle SOA composite application is deployed.
• Composite Name	The name of the Oracle SOA composite application in Oracle JDeveloper.
• Composite Version	The version of the Oracle SOA composite application in Oracle JDeveloper.
• Composite Instance ID	The Oracle SOA composite instance ID.
• Service Name	The name of the healthcare integration service binding component in the Oracle SOA composite application.
• Reference Name	The name of the Oracle Healthcare reference binding component in the Oracle SOA composite application.
Resubmit Count	The number times a message (to be displayed) has been resubmitted. Select from **Any**, **Equals**, **Not equal**, **Less than**, **Less than or equals**, or **Greater than**, or **Greater than or equals**, and then enter a defined count.
Last Resubmitted Date	The date when a message (to be displayed) was last resubmitted. Select from the **Any**, **Last**, or **Range** options.

4. To revert any unsaved changes you made to the report filters, click **Revert**.

5. When you are done specifying report filters, do any of the following:

- To test your filter criteria, click **Search**, and then click the **Show/Hide Filter** toggle button.

- To save the changes you made to the filter criteria for the report, click **Save AS New**.

Figure 9-12 displays a custom Application Message Report, which lists all the Application Messages exchanged in the last 30 days having state as Complete.

Figure 9-12   A Custom Application Message Report

You can sort the messages based on any of the report columns by clicking the column headers.

9.2.4 Specifying a Default Report

You can select one report that automatically appears when you click the Reports tab. This is the default report.

To specify a default report

1. On the Oracle SOA Suite for healthcare integration user interface, click the **Reports** tab. The Reports Welcome page appears when there are no opened reports.

2. Click the down arrow next to the **Selected Report** field, and select the report that you want to make default.

3. Click **Apply**.

   The selected report appears. The next time you select the **Reports** tab, this report automatically appears.

9.2.5 Configuring Reports

After you create a report, you can use the Message Report Filter Customizer to modify the filter criteria for the report. You can also create a report using an existing report as a template.

To configure a report

1. With the report you want to configure displayed on the Reports tab, click **Show Filter**.

   The Message Report Filter Customizer appears.

   Figure 9-13   Message Report Filter Customizer
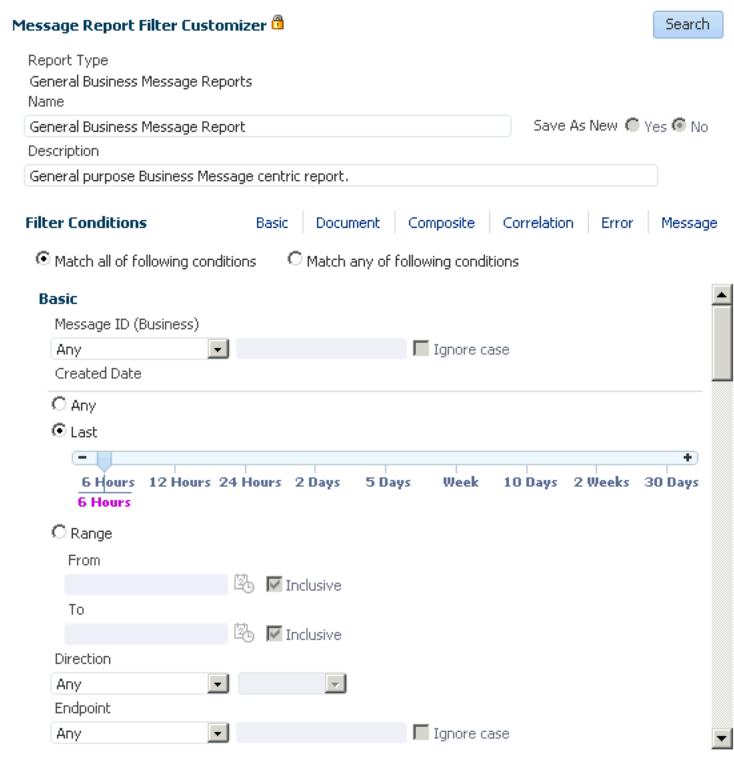
2. Modify any of the filters described in Table 9-1, Table 9-2, or Table 9-3 based on the report type.

3. To revert any unsaved changes you made to the report filters, click **Revert**.

4. When you are done specifying report filters, do any of the following:

   • To test your filter criteria, click **Search**, and then click the **Show/Hide Filter** toggle button.

   • To save the changes you made to the filter criteria for the report, click **Save As New**.

9.2.6 Refreshing a Report and Setting the Auto-Refresh Rate

You can manually refresh the displayed report at any time, but you can also specify that the report be automatically updated at set intervals. Note that the auto-refresh rate is only activated and configured for the current session. If you close and then re-open a report, the auto-refresh option is disabled.

To refresh a reports and set the auto-refresh rate

1. If the report you want to monitor is not already displayed, click the plus button in the upper right of the main Reports page and select the report to configure.

2. To refresh the report, click the **Refresh** button in the upper right.

3. To enable the auto-refresh option, do the following:

   a. Click the down arrow to the right of the **Refresh** button in the report toolbar.

   b. Specify the time interval in seconds to wait between automatically refreshing the report, and then select the **Auto-Refresh** check box.

   c. To disable automatic refreshing, clear the **Auto-Refresh** check box.

9.2.7 Deleting Reports

You can delete any of the user-created reports if you have the Administrator role. You can also delete any of the predefined reports, but use caution when doing this because this action is irreversible.

To delete a report

1. From the main Reports page, click the plus button in the upper right and select **Delete Reports**.

   The Delete Business Message Reports dialog appears.

   Figure 9-14   Delete Business Message Reports Dialog
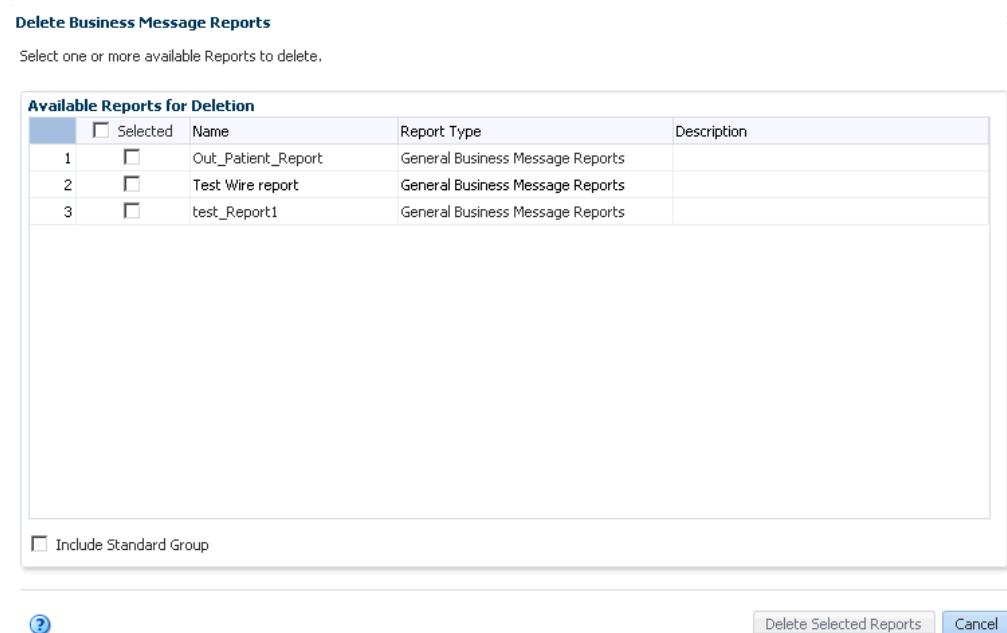
2. By default, only the user-created reports appear in the list. To include predefined reports, select **Include Standard Group**. This only pertains to standard reports imported from previous versions of the Oracle Healthcare console such as All, Last 24 Hours, Last 2 Weeks, and so on.

3. In the reports list, select the check box next to the reports you want to delete.

4. Click **Delete Selected Reports** or click **Cancel** to close the dialog box without deleting any reports.

9.3 Viewing Reports and Report Information

When you first open a report, a list of messages matching the report criteria appears on the Reports tab. This list shows summary information for each message.

The summary information includes the following:

- Message ID

- Created Date and Time

- Direction

- Endpoint

- State

- Document Type

You can select any of the messages in the list to view more detailed information and you can view a summary of multiple messages, as described in the following:

- "Viewing a Business Message Instance"

- "Viewing a Wire Message"

- "Viewing an Application Message"

- "Viewing the Flow Trace in Oracle Enterprise Manager"

- "Viewing Overview Information for Multiple Messages"

If you do not have permissions to view the document type of a message, certain details are hidden from view. The option to download information is disabled unless you have the required permissions.

9.3.1 Viewing a Business Message Instance

Business Messages include instance information for a document protocol, including the endpoint name, the message direction, the message ID, the state, the transport protocol and document protocol, and message details.

To view a business message instance

1. Display a report on the Reports page.

2. In the messages list, select the message for which you want to view instance data, then click the button highlighted in red in Figure 9-15 to navigate through the details of the business message. Refer to End-to-End Monitoring of Runtime Data for more information about the higher level reporting format.

Figure 9-15 Expand Details of the Business Message

The message flow diagram appears below the message list as shown in Figure 9-16.

Figure 9-16   Business Message Flow

3. To view the message details in a larger area, click **Business Message** in the message flow diagram and then click the **Collapse Pane** button beneath the message list. You can also click **Pin Current Message Details Into a New Tab** to open a new page containing the detailed information.

   The message details appear as shown in Figure 9-17.

Figure 9-17   Business Message Details

4. Expand or collapse sections of the business message by clicking the arrow button next to the section you want to expand or collapse.

5. To download information, click **Download As XML** or **Download as Text** in the section you want to download.

   > **Note:**
   >
   > This option is not available in all sections.

6. If you collapsed the message list pane, click **Restore Pane** above the message flow diagram to return to the message list.

9.3.2 Viewing a Wire Message

Wire messages are the native format of data sent into or out from an endpoint. Wire message details include message IDs, communication and protocol information (such as the transport binding and header details), payload message content, packed message content, and security information.

To view a wire message

1. Display a report on the Reports page.

2. In the messages list, select the message for which you want to view instance data.

   The message flow diagram appears below the message list as shown in Figure 9-18.

Figure 9-18   Wire Message Flow

3. To view the message details in a larger area, click **Wire Message** in the message flow diagram and then click the **Collapse Pane** button beneath the message list. You can also click **Pin Current Message Details Into a New Tab** to open a new page containing the detailed information.

   The message details appear as shown in Figure 9-19.

Figure 9-19   Wire Message Details

4. Expand or collapse sections of the Wire Message by clicking the arrow button next to the section you want to expand or collapse.

5. To download information, click **Download As XML** or **Download as Text** in the section you want to download.

Note:

This option is not available in all sections.

---

6. If you collapsed the message list pane, click **Restore Pane** above the message flow diagram to return to the message list.

9.3.3 Viewing an Application Message

Application messages provide information related to the SOA composite if a back-end composite application sent or received the message, including the composite name, version, instance ID, and so on.

To view an application message

1. Display a report on the Reports page.

2. In the messages list, select the message for which you want to view instance data.

The message flow diagram appears below the message list as shown in Figure 9-20.

Figure 9-20   Application Message Flow

3. To view the message details in a larger area, click **Application Message** in the message flow diagram and then click the **Collapse Pane** button beneath the message list. You can also click **Pin Current Message Details Into a New Tab** to open a new page containing the detailed information.

The message details appear as shown in Figure 9-21.

Figure 9-21   Application Message Details

4. Expand or collapse sections of the application message by clicking the arrow button next to the section you want to expand or collapse.

5. To download information, click **Download As XML** or **Download as Text** in the section you want to download.

Note:

This option is not available in all sections.

---

6. If you collapsed the message list pane, click **Restore Pane** above the message flow diagram to return to the message list.

9.3.4 Viewing the Flow Trace in Oracle Enterprise Manager

Oracle Healthcare reports provide links for each message to the flow trace of the composite in Oracle Enterprise Manager. Clicking the link automatically opens Oracle Enterprise Manager to the correct Flow Trace page.

To view the flow trace in Oracle Enterprise Manager

1. Display a report on the Reports page.

2. In the messages list, select the message for which you want to view the flow trace.

The message flow diagram appears below the message list.

Figure 9-22   Oracle Healthcare Message Flow

3. Click the composite link in the message flow diagram.

The Login page of Oracle Enterprise Manager opens in a separate browser tab or window.

4. Enter your login information for Oracle Enterprise Manager and then click **Login**.

The flow trace appears for the selected message as shown in Figure 9-23.

Figure 9-23   Message Flow Trace in Oracle Enterprise Manager

9.3.5 Viewing Overview Information for Multiple Messages

You can view a summary of information for multiple selected messages in the messages list of a report. This gives you a quick view of all selected messages, including the number of messages processed, the number of endpoints and document types used, the number of errors, the number of processed messages, and so on.

To see an overview of multiple messages

1. Display a report on the Reports page.

2. In the messages list, select the messages for which you want to view summary data by clicking the first message that you want to delete, press Shift, and then click the last message of the desired range.

   An overview of the selected messages appears beneath the list as shown in Figure 9-24.

Figure 9-24   Multiple Messages Selected for a Report

3. From here you can purge or resubmit the selected messages.

For more information, see "To resubmit multiple messages" or "To purge messages from the repository".

9.4 Working with Reports for Unassociated Messages

If a wire message or an application message does not have any associated business message, they are categorized as unassociated messages.

Unassociated messages can be viewed in Unassociated Wire Message reports and Unassociated Application Message reports.

9.4.1 Working with Unassociated Wire Messages

A burst of incoming messages from an external endpoint results in over-burdening the Oracle Healthcare runtime processing capability. These messages then gets temporarily backlogged and marked as PROCESSING. As a result of which, these Wire Messages do not have any associated Business Messages. Immediate ACKs generated by an inbound HL7 MLLP endpoint also results in Wire Messages without any associated Business Messages.

9.4.1.1 Creating Unassociated Wire Message Reports

You can create an Unassociated Wire Message report in a way similar to creating General Wire Message reports.

To create an unassociated wire message report

1. Perform Steps 1 and 2 from Creating Business Message Reports.

2. On the Create Message Report dialog, enter a name for the report, select **Unassociated Wire Message Reports** from the **Report Type** list, and click **Create**.

   The report appears with the Message Report Filter Customizer displayed.

3. On the Message Report Filter Customizer, enter any of the filter criteria described in Table 9-2.

   > **Note:**
   >
   > Multiple conditional operators are supported for most string criteria. For more information about using **Any**, **Equals**, **Like**, **Not Equal**, or **Not Like**, see About the Message Report Filter Customizers.

4. To revert any unsaved changes you made to the report filters, click **Revert**.

5. When you are done specifying report filters, do any of the following:

   - To test your filter criteria, click **Search**, and then click the **Show/Hide Filter** toggle button.

   - To save the changes you made to the filter criteria for the report, click **Save**.

Figure 9-25 displays an Unassociated Wire Message Report, which lists all the outbound unassociated Wire Messages in the last 24 hours.

Figure 9-25   An Unassociated Wire Message Report

You can sort the messages based on any of the report columns by clicking the column headers.

9.4.1.2 Viewing an Unassociated Wire Message

You can view an unassociated Wire Message from an Unassociated Wire Message report.

To view a wire message

1. Display an unassociated Wire Message report on the Reports page.

2. In the messages list, select the message for which you want to view instance data.

    The message flow diagram appears below the message list as shown in Figure 9-26.

Figure 9-26   Unassociated Wire Message Flow

> **Note:**
>
> The unavailable Business or Application Message icons are rendered as disabled in the message flow.

3. To view the message details in a larger area, click **Wire Message** in the message flow diagram and then click the **Collapse Pane** button beneath the message list. You can also click **Pin Current Message Details Into a New Tab** to open a new page containing the detailed information.

4. Expand or collapse sections of the unassociated Wire Message by clicking the arrow button next to the section you want to expand or collapse.

5. To download information, click **Download As XML** or **Download as Text** in the section you want to download. However, because the message does not have any associated Business Message, Payload does not contain any data.

> **Note:**
>
> This option is not available in all sections.

6. If you collapsed the message list pane, click **Restore Pane** above the message flow diagram to return to the message list.

9.4.2 Working with Unassociated Application Messages

A burst of Application Messages emitted by Fabric/Composite layer results in over-burdening the Oracle Healthcare runtime processing capacity. These messages then gets temporarily backlogged and marked as PROCESSING. As a result of which, these Application Messages do not have any associating Business Messages. Severe errors encountered by an outbound message from Fabric/Composite layer also results in Application messages without associated Business Messages.

9.4.2.1 Creating Unassociated Application Message Reports

You can create an Unassociated Application Message report in a way similar to creating General Application Message reports.

To create an unassociated application message report

1. Perform Steps 1 and 2 from Creating Business Message Reports.

2. On the Create Message Report dialog, enter a name for the report, select **Unassociated Application Message Reports** from the **Report Type** list, and click **Create**.

   The report appears with the Message Report Filter Customizer displayed.

3. On the Message Report Filter Customizer, enter any of the filter criteria described in Table 9-3.

> **Note:**
>
> Multiple conditional operators are supported for most string criteria. For more information about using **Any**, **Equals**, **Like**, **Not Equal**, or **Not Like**, see About the Message Report Filter Customizers.

4. To revert any unsaved changes you made to the report filters, click **Revert**.

5. When you are done specifying report filters, do any of the following:

- To test your filter criteria, click **Search**, and then click the **Show/Hide Filter** toggle button.

- To save the changes you made to the filter criteria for the report, click **Save**.

[Reviewer: Please provide a sample Unassociated Application Message report image.]

You can sort the messages based on any of the report columns by clicking the column headers.

9.4.2.2 Viewing an Unassociated Application Message

You can view an unassociated Application Message from an Unassociated Application Message report.

To view an application message

1. Display an unassociated Application Message report on the Reports page.

2. In the messages list, select the message for which you want to view instance data.

The message flow diagram appears below the message list as shown in.

[Reviewer: Please provide an image of a sample Unassociated Application Message flow]

> **Note:**
>
> The unavailable Business Message button is rendered as disabled in the message flow.

3. To view the message details in a larger area, click **Application Message** in the message flow diagram and then click the **Collapse Pane** button beneath the message list. You can also click **Pin Current Message Details Into a New Tab** to open a new page containing the detailed information.

4. Expand or collapse sections of the unassociated Wire Message by clicking the arrow button next to the section you want to expand or collapse.

5. To download information, click **Download As XML** or **Download as Text** in the section you want to download. However, because the message does not have any associated Business Message, Payload does not contain any data.

> **Note:**
>
> This option is not available in all sections.

6. If you collapsed the message list pane, click **Restore Pane** above the message flow diagram to return to the message list.

9.5 Working with Error Messages

If processing for a message results in an error, the message appears in the message list with a red box over the icon.

You can select error message to view additional information about the error.

9.5.1 Viewing an Error Message

This report provides information related to errors in message processing, including the error message, code, severity, level, and description.

To view an error message

1. Review the messages list to find messages with errors, as indicated by an error icon or a state of MSG_ERROR as shown in Figure 9-27.

Figure 9-27   Error Message in Messages List

	Message ID	Created Date and Time	Direction	Endpoint	State
1	0AB16502145508B7137000004D06C865-1	Apr 11, 2014 4:58 PM	INBOUND	Emergency_Laboratory	MSG_COMPLETE
2	0AB165021450DC07B0C000004D06C85A-1	Mar 29, 2014 5:41 PM	INBOUND	Emergency_Laboratory	MSG_COMPLETE
3	0AB165021450DC0774B000004D06C83B-1	Mar 29, 2014 5:41 PM	INBOUND	Emergency_Admission	MSG_COMPLETE
4	0AB165021450DC077BB000004D06C846	Mar 29, 2014 5:41 PM	OUTBOUND	Emergency_Admission	MSG_ERROR
5	0AB165021450DC07782000004D06C843	Mar 29, 2014 5:41 PM	OUTBOUND	Emergency_Laboratory	MSG_COMPLETE
6	0AB165021450D27DB87000004D06C81B	Mar 29, 2014 2:54 PM	OUTBOUND	Emergency_Admission	MSG_COMPLETE
7	0AB165021450D27DB58000004D06C818	Mar 29, 2014 2:54 PM	OUTBOUND	Emergency_Laboratory	MSG_COMPLETE
8	0AB165021450D27DF16000004D06C82F-1	Mar 29, 2014 2:54 PM	INBOUND	Emergency_Laboratory	MSG_COMPLETE
9	0AB165021450D27DB06000004D06C810-1	Mar 29, 2014 2:54 PM	INBOUND	Emergency_Admission	MSG_COMPLETE
10	0AB165021450C8F3901000004D06C7F0	Mar 29, 2014 12:08 PM	OUTBOUND	Emergency_Admission	MSG_COMPLETE
11	0AB165021450C8F38D4000004D06C7ED	Mar 29, 2014 12:08 PM	OUTBOUND	Emergency_Laboratory	MSG_COMPLETE
12	0AB165021450C8F3C7A000004D06C804-1	Mar 29, 2014 12:08 PM	INBOUND	Emergency_Laboratory	MSG_COMPLETE

Found 421 results.

2. When you find an error message to view, select the message.

   The message flow diagram changes to display where the error occurred as shown in Figure 9-28.

Figure 9-28   Message Flow Diagram With an Error

3. Click the error button in the message flow diagram.

   The error message details appear beneath the diagram as shown in Figure 9-29.

Figure 9-29   Error Message Details

Error Details

Error Text	Transport error: Message was not sent - error
Error Code	HC-50079
Error Severity	ERROR
Error Level	ERROR_LEVEL_COLLABORATION
Error Description	Machine Info: (blr2261789) Transport error: Message was not sent - error

9.5.2 Resubmitting Messages

You can resubmit Wire and Application Messages from the Reports tab. When a message transaction contains an error, use the reporting features to determine the nature of the error and correct it. After the issue is fixed, resubmit the message. You can resubmit messages one at a time or in a group.

To resubmit a message

1. In the messages list for the displayed report, select the message you want to resubmit.

2. Do one of the following:

   • Select **Resubmit**, and then select **Associated Application Message** or **Associated Wire Message**.

   • Right-click the selected message and use the context menu items.

   • In the message flow diagram, right-click either **Wire Message** or **Application Message**, and then select **Resubmit Wire Message** or **Resubmit Application Message**.

3. On the confirmation dialog that appears, click **OK**.

   The error icon on the message in the messages list changes to a green arrow to indicate the message was resubmitted.

To resubmit multiple messages

1. In the messages list for the displayed report, select the messages you want to resubmit.

   A summary of information about the selected messages appears beneath the messages list as shown in Figure 9-30.

Figure 9-30   Multiple Messages Selected on the Reports Page

2. In the yellow bar above the Selection Overview, click **Resubmit**, and then select **Associated Application Message** or **Associated Wire Message**. You can also use the **Resubmit** menu on the toolbar or right-click the selected message and use the context menu items.

3. On the confirmation dialog that appears, click **OK**.

9.6 Purging Messages from the Repository

You can purge messages that you no longer want to store in the healthcare integration repository. If a message you purge is part of a batch process, all messages in the batch are also deleted.

You can purge messages one at a time or as a group.

To purge messages from the repository

1. In the messages list of the displayed report, select the message or messages you want to remove.

2. Click **Purge**.

3. On the confirmation dialog that appears, click **Yes**.

Note:

When you a Business Message, the corresponding entry in the Sequence Manager table also gets deleted.

9.7 End-to-End Monitoring of Runtime Data

In an HL7 interface engine implementation that uses Oracle SOA Suite for healthcare integration, the reports page has the capability to associate the source HL7 endpoint data to multiple targets.

This is shown in Figure 9-31.

Figure 9-31   General Business Message Report Page
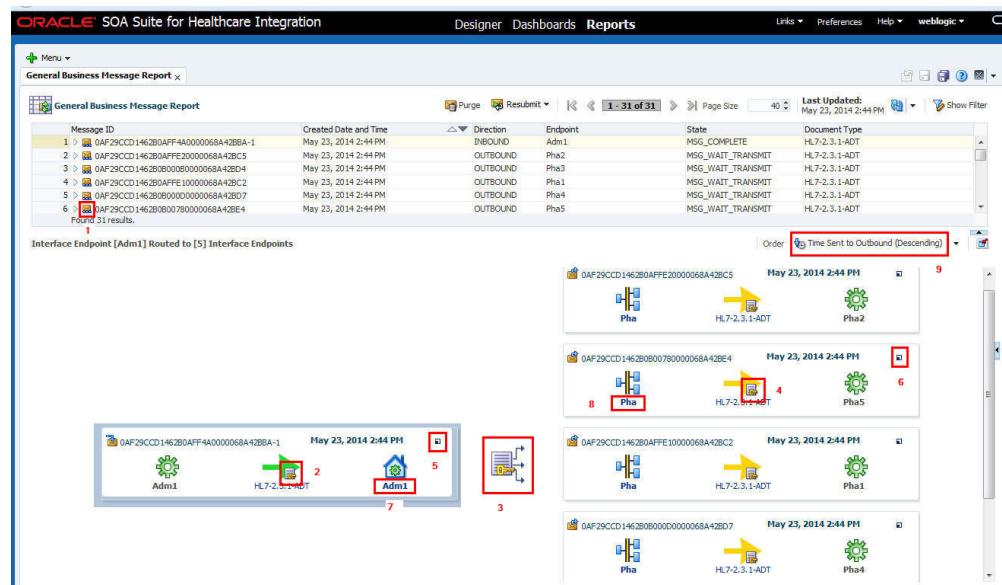
Figure 9-31 shows the source HL7 endpoint `Adm1` fanning out to five different targets (`Pha1-5`). As described in Viewing a Business Message Instance, you can view the business message details by clicking the button labeled "5" in Figure 9-31. This source to target(s) view is displayed only if you select messages with the icon labeled "1". Selecting messages without the icon displays the business message details.

Click the button labeled "2" in Figure 9-31 to display the HL7 native data. The green arrow indicates that the segment is completed. A yellow arrow indicates the segment is in progress and a red arrow indicates the segment is in error. In Figure 9-31, the target endpoints are shown in the pending state. When successfully completed, the color of the arrows changes to green. The label beneath the arrow shows the document type and the version of HL7 messages being processed.

Click the button labeled "3" in Figure 9-31 to display the XML data of the endpoint that is in focus. For example, Figure 9-31 shows that the source endpoint `Adm1` is in focus. Click the button labeled "3" to display the XML data from the source endpoint `Adm1` (assuming translation is enabled). This XML data passes to the composite for routing and transformation. If the endpoint in focus is a target endpoint, click the button labeled "3" to display the XML data coming from the composite. Clicking the button labeled "3" results in different XML data values (if translation is enabled) depending on which endpoint is in focus. Also, note that the icons labeled "2", "3" and "4" have the image of a key embedded in them. The key indicates that if auditing functionality is

turned on to the payload, the act of clicking these icons is logged for auditing purposes.

The icons labeled "7" and "8" indicate the internal delivery channels used to communicate to and from the composite; they can be JMS queue names or a composite. These icons cannot be selected and are meant to convey how these messages are passed to and from the composite.

By default, the icon labeled "9" shows the order of the target endpoints determined in descending order by the time sent. The order can be changed with other options such as "Time Sent to Outbound (Ascending)", "Outbound Endpoint Name (Ascending)" or "Outbound Endpoint Name (Descending)" This feature is useful if there are multiple target endpoints and you want to determine if the target endpoints are processed. If the HL7 message is not passed from the composite to the target endpoint, the endpoint is not displayed in this report page.

Note:

By default, the outbound messages are shown as flowing from left to right. You can use the User Interface Settings page (Administration > Settings > UI) to change the flow to be right to left by setting **Show Outbound Message Flow in Right-to-Left Order** to Yes.

10

Configuring Alerts and Contacts

This chapter describes how to define alert notifications and contacts in Oracle SOA Suite for healthcare integration. You can specify that certain people are alerted by email or text (SMS) when certain healthcare-related runtime or design-time events occur.

This chapter includes the following topics:

- Overview of Alerts and Contacts
- Configuring Contacts and Alerts
- Viewing the Alerts Assigned to a Contact
- Removing Contacts
- Viewing a History of Alerts Sent

10.1 Overview of Alerts and Contacts

In healthcare integration, administrators and support personnel must know when certain events occur, such as changes to the healthcare configuration or errors that affect processing. Oracle SOA Suite for healthcare integration addresses this by allowing you to define contact information and associate each contact with specific events for which alerts are generated.

For example, if an error occurs during message processing or delivery, an email or text alert can be sent to the appropriate people so the issue can be handled immediately.

The alert notification feature for healthcare integration works in conjunction with the built-in notification framework of Oracle User Messaging Service (UMS) and Oracle SOA Suite human workflow services.

10.2 Configuring Contacts and Alerts

Contacts and alerts are configured using the Oracle WebLogic Administration Console, Oracle Enterprise Manager, and the healthcare integration user interface. You configure the email server and sender address information, as well as enable all notifications, from Oracle Enterprise Manager. You define the individual contacts and subscribe them to alerts using the healthcare integration user interface.

Configuring contacts and alerts involves the following steps:

- Deploying the SMPP Driver for SMS Notifications
- Configuring Workflow Notification Properties
- Configuring Oracle User Messaging Service
- Defining Alerts and Contacts

For information about configuring User Messaging Service, see `Oracle® Fusion Middleware Administering Oracle User Messaging Service`.

10.2.1 Deploying the SMPP Driver for SMS Notifications

If you are using SMS (text) notifications, you must specify the target servers for the SMPP driver for the User Messaging Service. Perform this task on the Oracle WebLogic Administration Console.

To deploy the SMPP driver

1. In a web browser, launch Oracle WebLogic Administration Console and log in. The URL is:

   `http://hostname:port_number/console`

   The hostname is the name of the computer on which WebLogic Server resides, and the port number is the port on which WebLogic Server listens (by default, 7001).

2. In the Domain Structure navigator, click Deployments.

3. In the Deployments list that appears, select **usermessagingdriver-smpp**.

   The Settings for usermessagingdriver-smpp page appears.

4. Click the **Targets** tab.

   The Target Assignments list appears.

Figure 10-1   Targets Tab for usermessagingdriver-smpp Settings

5. In the Target Assignments table, select the check box next to **Component** to select all driver components, and then click **Change Targets**.

   The Change Targets Assistant appears.

Figure 10-2   Target Deployments for usermessagingdriver-smpp
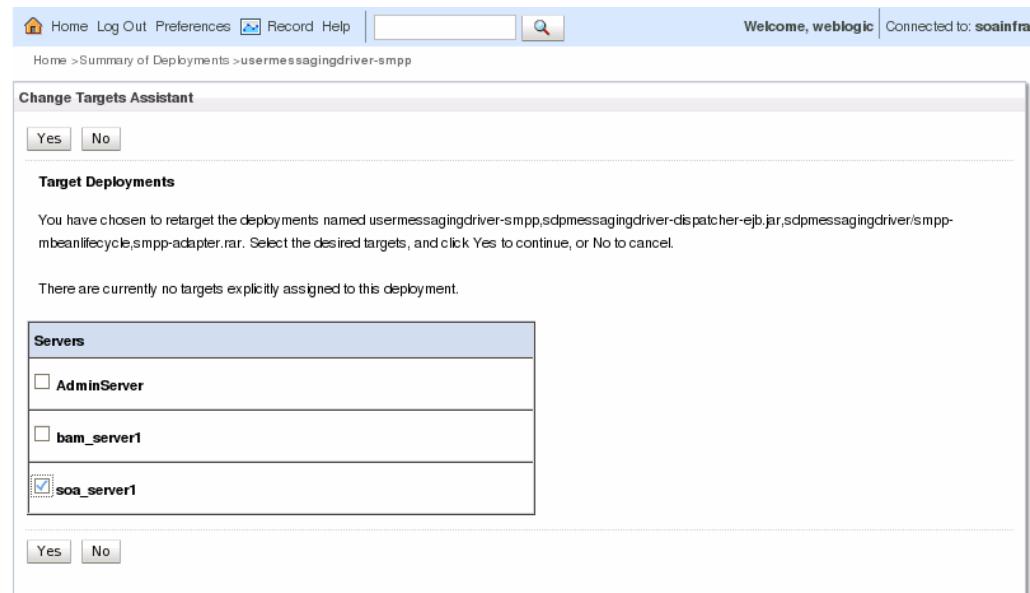
6. Select the SOA Manager Server you are using for healthcare integration, and then click **Yes**.

Confirmation messages appear at the top of the page letting you know whether the targets were changed successfully.

10.2.2 Configuring Workflow Notification Properties

Workflow notification properties enable the notification feature for all notification types, and also define email addresses for the sending email account for the notifications.

To configure the workflow notification properties

1. In a web browser, launch Oracle Enterprise Manager and log in. The URL is:

```
http://hostname:port_number/em
```

The hostname is the name of the computer on which WebLogic Server resides, and the port number is the port on which WebLogic Server listens (by default, 7001).

2. In the left navigation panel, expand **SOA** and select **soa-infra**.

3. On the soa-infra page, click the **SOA Infrastructure** menu, point to **SOA Administration**, and select **Workflow Properties**.

4. On the Workflow Notification Properties page, set the **Notification Mode** to **All**.

5. In the Notification Service section, configure the following email addresses:

- **Email: From Address** -- The email address from which you want to send email notifications.

- **Email: Actionable Address** -- The email address for actionable incoming messages. This is not required.

- **Email: Reply To Address** -- The email address to include in the Reply To field of the outgoing email notification header.

Figure 10-3   SOA Infrastructure Workflow Notification Properties
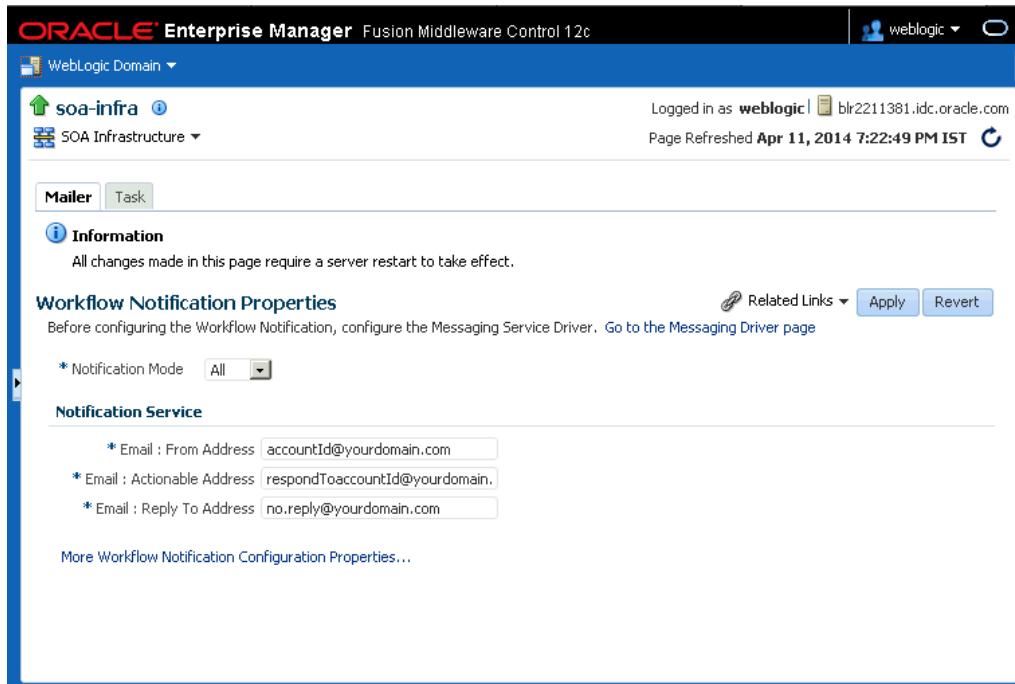
6. Click **Apply**, and then click **Yes** on the confirmation dialog.

Note:

Whenever you add a document definition to an endpoint, the back-end process of Oracle SOA Suite for healthcare integration deploys the document. So, an email is sent for the document deployment, and no notification is sent for the document addition.

10.2.3 Configuring Oracle User Messaging Service

Oracle UMS provides the underlying infrastructure for notifications. In order to use email notifications, you must configure email driver information for the UMS; in order to use SMS (text) notifications, you must configure SMPP driver information.

To configure Oracle User Messaging Service

Note:

For more information about this task and the properties you can configure, see `Oracle® Fusion Middleware Administering Oracle User Messaging Service`.

---

1. In the left navigation panel of Oracle Enterprise Manager, expand **User Messaging Service**.

2. Right-click **usermessagingdriver-email** and select **Email Driver Properties**.

Figure 10-4   Email Driver Properties Page for UMS
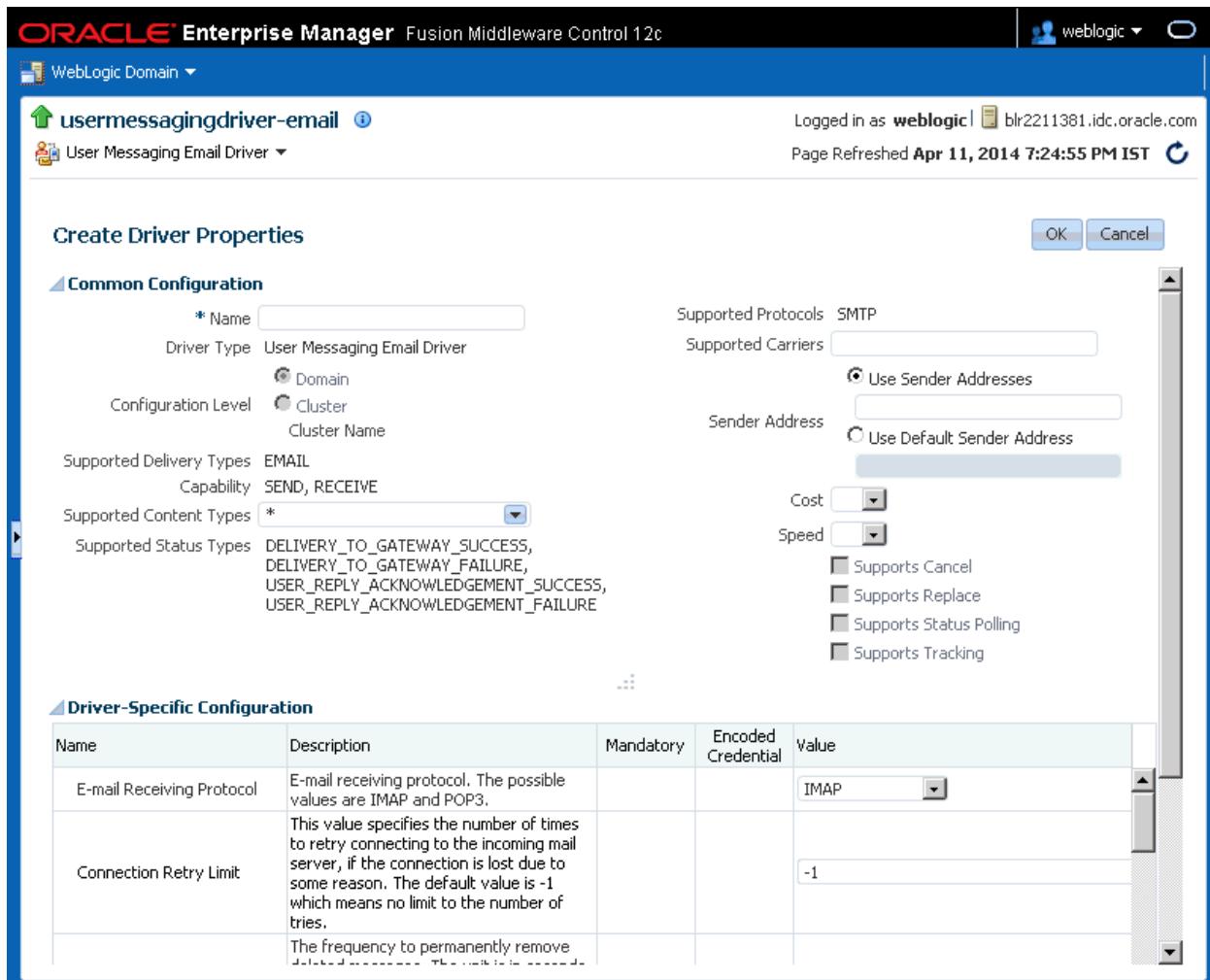
3. Configure the email driver properties for your email server.

4. After you configure the email properties, right-click **usermessagingdriver-smpp** under **User Messaging Service** in the left navigation pain

5. Select **SMPP Driver Properties**.

6. Configure the SMPP driver properties.

7. After you configure all the workflow and email driver properties, restart the server to load all the changes.

10.2.4 Defining Alerts and Contacts

In the healthcare integration user interface, you can define contacts and then subscribe each contact to multiple alerts for different types of events. The events that can generate an alert are listed in Table 10-1.

To define alerts and contacts

1. In a web browser, launch the Oracle SOA Suite for healthcare integration user interface. The URL is:

   `http://host_name:port_number/healthcare`

   The hostname is the name of the computer on which the WebLogic Managed Server resides; and port number is the port number on which the Managed Server listens.

2. Under the **Designer** tab, click the **Administration** tab and then click **Alerts and Contacts**.

Figure 10-5   Alerts and Contacts Page
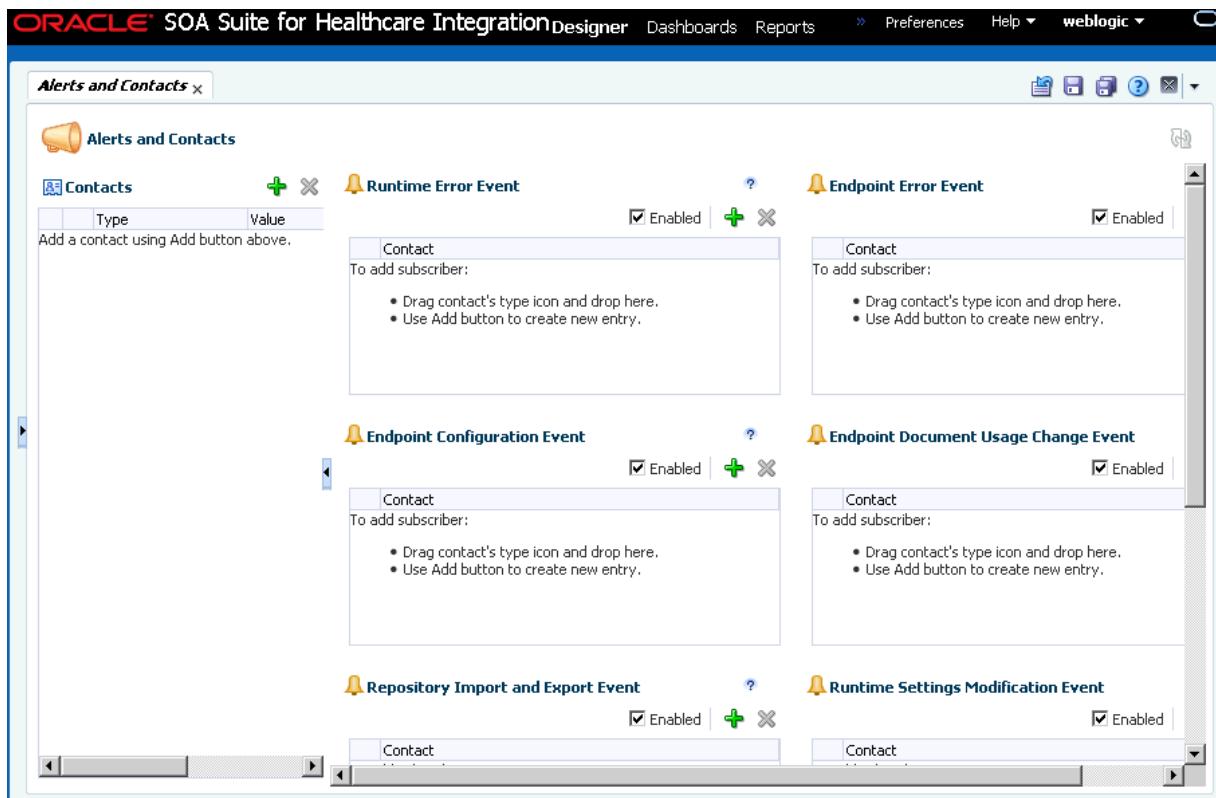
3. In the Contacts section, click **Add a Contact** (the plus button in the upper right).

4. Enter the following information:

   - **Type**: Select the type of alert to send. Choose from **Email** or **Texting (SMS)**.

   - **Value**: The email address or text number to which alerts should be sent. Make sure the type of value you enter here corresponds to the type you selected.

5. Repeat the previous two steps for each contact you must add.

Figure 10-6   Texting and email Contacts

6. Click **Save** to save the contact information.

7. In the Events section on the right, select the **Enabled** check box for all events for which you want to send alerts.

8. To subscribe contacts to events, do one of the following:

    *   For each contact you create, drag and drop the contact from the contacts list into one or more of the enabled events listed on the right side. Table 10-1 lists and describes each event for which you can configure alerts.

    *   Click **Add an Alert for this Event** next to the event name, and then select a contact from the drop down list that appears in the event section.

Figure 10-7   Alert Events With Contacts Assigned

Each time an event occurs, the specified message type (email or text) is sent to the contacts listed for that event.

> **Tip:**
>
> Due to the length of some runtime error messages, you might want to avoid subscribing text (SMS) contacts to runtime events.

Table 10-1   Healthcare Integration Alert Events

Table 10-1   (Cont.) Healthcare Integration Alert Events

Event	Description
Runtime Error	Triggers an alert when an error occurs during inbound or outbound message processing, delivery to endpoints, or delivery to internal delivery channels.
Endpoint Error	Triggers an alert when an error occurs in the endpoint transport layer; for example, the TCP/IP server port is already in use.
Endpoint Configuration	Triggers an alert when changes are made to an endpoint's configuration, including enabling and disabling an endpoint.
Endpoint Document Usage Change	Triggers an alert when changes are made for a document definition associated with an endpoint.
Repository Import and Export	Triggers an alert when data is imported into or exported from the healthcare integration repository.
Runtime Settings Modification	Triggers an alert when changes are made to the any of the runtime properties listed on the Runtime Settings page.
Purge Events	Triggers an alert when: • The runtime data is purged with **Purge Control Number** selected or deselected under **Repository Management** in the **Administration** tab. • The runtime data is purged from the command line with the `purgecontrolnumber` option set to `true`.

10.3 Viewing the Alerts Assigned to a Contact

You can view all of the alerts that are assigned to a specific contact from the Contacts list on the Alerts and Contacts page.

To view alerts for a contact

1. Under the **Designer** tab of the healthcare integration user interface, click the **Administration** tab and then double-click **Alerts and Contacts**.

2. In the **Contacts** list, click the arrowhead button to the left of the contact whose alert information you want to view.

   A list of alert subscriptions appears.

Figure 10-8   Alert Subscriptions for a Contact
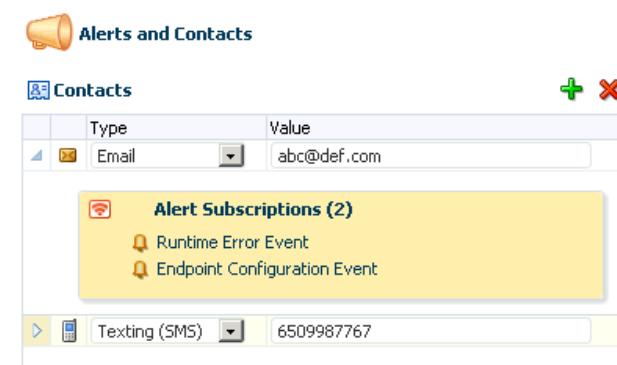
3. When you are done viewing the alerts, click the contact's arrowhead again to close the alert subscriptions list.

10.4 Removing Contacts

After you create a contact, you can remove it from the contact list if it is no longer required. If a contact no longer must receive alerts for an event, you can unsubscribe the contact from that event.

To Remove a Contact

1. Under the **Designer** tab of the healthcare integration user interface, click the **Administration** tab and then double-click **Alerts and Contacts**.

2. In the Contacts list, select the contact to delete, and then click **Delete Currently Selected Contact** (the red X in the upper right of the Contacts list).

3. If the contact is subscribed to any alerts, a confirmation dialog appears. Click **OK** to remove the contact.

   The contact is removed from the Contacts list and is also removed from any events to which it was subscribed.

To Unsubscribe a Contact from an Alert

1. Under the **Designer** tab of the healthcare integration user interface, click the **Administration** tab and then double-click **Alerts and Contacts**.

2. Select the contact in the event list from which you want to remove the contact.

3. Click **Delete Currently Selected Alert** (the red X in the upper right of the event list.).

   The contact is removed from the list.

10.5 Viewing a History of Alerts Sent

You can view a history of the alerts that have been sent to each contact from the Oracle Enterprise Manager Human Workflow Engine pages.

To view a history of alerts sent

1. In a web browser, launch Oracle Enterprise Manager and log in. The URL is:

```
http://hostname:port_number/em
```

The hostname is the name of the computer on which WebLogic Server resides, and the port number is the port on which WebLogic Server listens (by default, 7001).

2. In the left navigation panel, expand **SOA** and select **soa-infra**.

3. On the soa-infra page, click the **SOA Infrastructure** menu, point to **Service Engines**, and select **Human Workflow**.

4. On the Human Workflow Engine page, click the **Notification Management** tab.

11

Viewing the Healthcare User Audit Trail

This chapter describes how to enable and configure an audit trail of user activity for healthcare integration components and applications. Oracle SOA Suite for healthcare integration uses Oracle's Common Audit Framework to log user activity against healthcare integration components.

This chapter contains the following topics:

- Introduction to the Audit Trail

- Configuring the Healthcare Integration Audit Trail

- Viewing User Audit Logs

11.1 Introduction to the Audit Trail

The Oracle auditing framework collects and stores information about events affecting configured components, providing an audit log of activity for those components to help support your compliance requirements. Auditing for each SOA Suite component is defined by an *audit policy* that defines which components and which activities are captured in the audit log.

You can configure the audit policy to only capture the information you require and ignore the rest. This is done on the Audit Policy page of Oracle Enterprise Manager. See "Managing Audit Policies" in the *Oracle Fusion Middleware Application Security Guide* for more information.

The set of auditable events for each application and component is defined by the audit policy and differs between each application. When you expand the list of events for a component, only those events that can be audited for that component appear in the list. For each event, you can further specify whether to only log successful attempts or failed attempts (currently Oracle SOA Suite for healthcare integration only logs successful attempts).

When you configure auditing, you can select from the following audit levels:

- **Low**: This option selects a subset of events from all auditable components in the audit policy list, including a subset of Oracle SOA Suite for healthcare integration events. It does not allow custom filters to be created.

- **Medium**: This option selects a larger subset of events from all auditable components in the audit policy list, including all Oracle SOA Suite for healthcare integration events. It does not allow custom filters to be created.

- **Custom**: This options lets you select only those components, events, and conditions that you want to audit. This is the recommended level for Oracle SOA Suite for healthcare integration. You must select this level in the Oracle Enterprise Manager console to enable Oracle Healthcare auditing.

You can also specify a list of users whose activity is audited regardless of the actions performed or the component used. Auditing occurs for these users no matter what audit level or filters are defined.

For more information about audit policies, see `Configuring and Managing Auditing` in the *Oracle Fusion Middleware Application Security Guide*.

11.1.1 Oracle SOA Suite for Healthcare Integration Auditing Options

The components and events available for auditing are listed on the Audit Policy page of Oracle Enterprise Manager (Weblogic domain > **Security** > **Audit Policy**). To view or configure the these options, select **Oracle SOA Suite for healthcare integration** from the **Audit Component Name** list, **Custom** from the **Audit Level** list, and click the check boxes adjacent to the events.

Figure 11-1   Healthcare Integration Components on the Audit Policy Page

> **Note:**
>
> Currently only the SUCCESS events are audited. You should not select FAILURE events.

Currently, the following components and events are supported for audit in Oracle SOA Suite for healthcare integration (note that additional events appear in the list, but they are not currently logged):

- User Session

  – User Login

- – User Logout

- Endpoint Management

  - – Enable Endpoint

  - – Disable Endpoint

- Document Management

  - – Resubmit Message

  - – Purge Message

  - – Read Payload

- Configuration

  - – Import

  - – Export

11.1.2 Using Filter Conditions for Auditing

For each event, you can define filters for the success condition. Filters use rule-based expressions that are based on the attributes of the event. For most Oracle SOA Suite for healthcare integration user access auditing, you can use the following attributes in your filter expressions:

- Host ID

- Host Network Address

- Initiator

- Client IP Address

- Resource

- Domain Name

- Target User

- Roles

- Audit User

Expressions can include AND and OR operators, as well as a variety of comparison functions, such as equals, starts with, contains, does not equal, and so on.

11.2 Configuring the Healthcare Integration Audit Trail

You configure audit policies in Oracle Enterprise Manager by selecting the events or components to include in the audit log. Currently, Oracle B2B components and events are not included in the audit trail.

There are two default configurations, Low and Medium audit levels, that select a predefined subset of components or events. These are not recommended for Oracle SOA Suite for healthcare integration because they affect all auditable components, not just the components of Oracle SOA Suite for healthcare integration. Selecting either of

these options can result in extraneous audit entries and unnecessarily large audit logs. Additionally, these two options do not allow you to define any filters.

The following instructions apply to custom-level audit policy configuration.

To configure auditing for healthcare integration

1. Login to Oracle Enterprise Manager.

2. In the navigation panel on the left, expand **WebLogic Domain** and then right-click the name of the domain for which you want to enable user auditing.

3. In the context menu that appears, point to **Security** and then select **Audit Policy**.

Figure 11-2 Security Context Menu for a WebLogic Domain
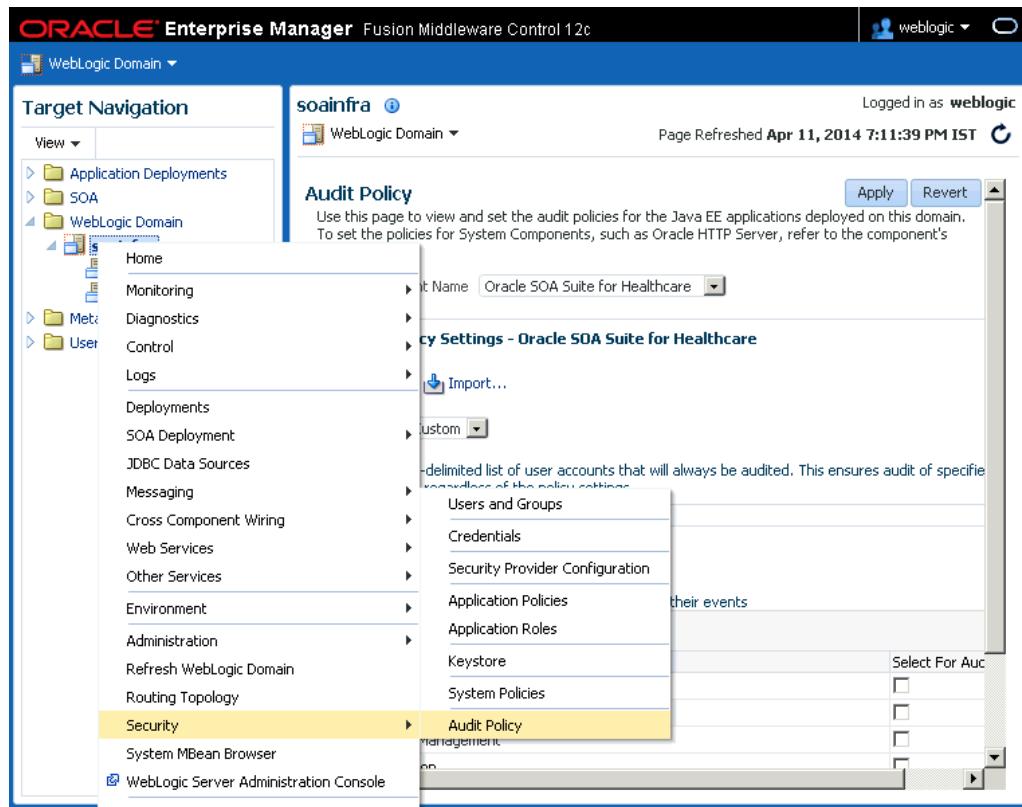
4. In the **Audit Component Name** list, select **Oracle SOA Suite for Healthcare**.

5. In the **Audit Level** list, select **Custom**.

   Check boxes appear in the **Select for Audit** column so you can select which healthcare integration components and events to audit.

6. Click event categories such as User Session to display the list of events pertaining to that category below.

7. Do any of the following:

   - To enable auditing for all Oracle SOA Suite for healthcare integration components and events, click the **Audit All Events** button.

- To enable auditing for all events for a specific component, select the check box in the **Select for Audit** column next to the component name.

  For example, to audit all actions taken against endpoints, select the check box for **Endpoint Management**.

Figure 11-3   Endpoint Management Component With All Events Selected

- To enable auditing of a specific event for a component, expand the component and select the check box in the **Enable Audit** column next to the event name under that component.

8. To define a filter for a success condition, select **Enable Audit** for the success condition, and then click its **Edit Filter** button. Define the filter on the dialog that appears, and then click **OK**.

    For more information about filters, see Using Filter Conditions for Auditing and the online help available from the Edit Filter dialog. Note that filters can only be defined for success conditions at this time.

9. To specify a list of users whose activity is always audited regardless of the component configuration, enter a list of user accounts in the **Users to Always Audit** section. Separate the account names with commas.

10. When you are done configuring auditing, click **Apply**.

11. Restart the server in order for the changes to take effect.

11.3 Viewing User Audit Logs

When an event triggers an audit log entry, the event information is written to the audit log file.

The audit log captures the following information. Depending on the type of event that triggered the entry, several of these fields might be empty.

- Date and time

- Initiator of the event

- Event type

- Event status

- Message text (indicating what occurred)

- ECID

- RID

- Context fields

- Session ID

- Target component type

- Application name

- Event category

- Thread ID

- Failure code

- Remote IP address

- Target

- Resource

- Roles

- Authentication method

- Reason

You can view the audit log file directly. It is written to the following location:

```
fmw_home/user_projects/domains/domain_name/servers/
managed_server_name/logs/auditlogs/SOA-HCFP/audit.log
```

12

Managing the Repository

This chapter describes the healthcare integration metadata repository and provides instructions for managing the stored data. Design-time and instance data for the Oracle SOA Suite for healthcare integration are stored in a metadata repository, and you can import data, export data, and purge metadata or instance data.

This chapter includes the following topics:

- Introduction to the Oracle SOA Suite for Healthcare Integration Repository

- Importing and Exporting the Design-Time Repository

- Purging Repository Data

12.1 Introduction to the Oracle SOA Suite for Healthcare Integration Repository

Oracle SOA Suite for healthcare integration instance data is stored and managed within the SOAINFRA schema of your database. Metadata for design-time and configuration is stored and managed through Metadata Services (MDS), available in Oracle Fusion Middleware.

For more information about MDS, see the *Oracle Fusion Middleware Administrator's Guide*.

12.1.1 Repository Maintenance

The healthcare integration user interface provides features to help you manage the repository data, which include importing repository data, exporting the full repository, purging design-time metadata, and purging instance data. You can import objects such as document definitions, map files, endpoints, an exported repository, and so on.

Design-time metadata includes endpoints, document definitions, internal delivery channels, and mapsets. Instance metadata is the information that is created during runtime when messages are processed. In addition to being able to purge these two types of data on the Repository Management page, you can also purge messages on the Reports page.

Use caution when using the import and purge features. During an import, you might overwrite existing data, and purging data removes the data permanently.

12.1.2 What Occurs During the Import or Export Process

Exporting a file exports the full healthcare integration metadata repository and creates a ZIP file containing the exported data. When you import a file, all of the objects in the export file is copied, which can include documents, endpoints, callouts, mapsets, and so on. If you choose to replace existing metadata during an import procedure, any

existing metadata with the same name as metadata in the export file is overwritten by the information being imported.

Note that library JAR files used by Java callouts are not copied during an import or export procedure.

When you export the design-time repository, continue to make changes to the repository contents in the healthcare integration user interface, and later import the exported file (the contents of which are now older), then updates occur as follows:

- If **Replace Existing Metadata** is *not* checked during import, then any new data that was created or modified in the healthcare integration user interface after the file was exported is left untouched.

- If **Replace Existing Metadata** is checked during import, then the existing metadata is replaced with the ZIP file metadata.

If an import fails, then the changes are rolled back and the design-time repository remains unchanged. A message appears indicating that the import was unsuccessful.

12.1.3 About the Exported File

Design-time repository contents that are exported to a file represent a copy of the current data. This file is no longer accessible for changes from the healthcare integration user interface until it is imported back into Oracle SOA Suite for healthcare integration. Do not manually edit exported files.

12.1.4 What Occurs During the Purging Process

You can purge both design-time metadata and instance data. Design-time metadata includes document definitions, endpoints, callouts, mapsets, and internal delivery channels. When you purge this data, predefined data that is part of the installation is not purged. After you perform a successful purge of design-time metadata, you are logged out of the healthcare integration user interface so you must log back in. You cannot purge design-time metadata if there are active endpoints.

Instance data is generated during runtime when messages are processed. Instance, or runtime, data contains the business messages and message-related data. Specific instance data can be purged from the Reports page. See "Working with Reports" for more information.

Purging is useful for:

- Managing disk space and improving performance

- Removing repositories on a test system

12.1.5 Purging Control Numbers

When you purge instance data, you can optionally purge control number information. Control numbers are used in HL7 message standards. Oracle SOA Suite for healthcare integration keeps track of control numbers for inbound and outbound messages. For outbound messages, the control numbers are generated in a sequence from an internal control number table. Because purging instance data and control numbers resets the sequence (the control number table is reset), an outbound message after a purge might have the same control number as a message before the purge. If this is undesirable, do not purge control numbers.

12.2 Importing and Exporting the Design-Time Repository

Oracle SOA Suite for healthcare integration design-time data can be exported and saved to a ZIP file. The ZIP file can be imported back into Oracle SOA Suite for healthcare integration so the data is available in the healthcare integration user interface. This is useful when migrating data from a test environment to a production environment.

> **Caution:**
>
> Do *not* manually edit exported files. This could make the data unstable after it is imported back in.

Figure 12-1 shows the Repository Management tab, where you import and export design-time data.

Figure 12-1   Repository Management Page
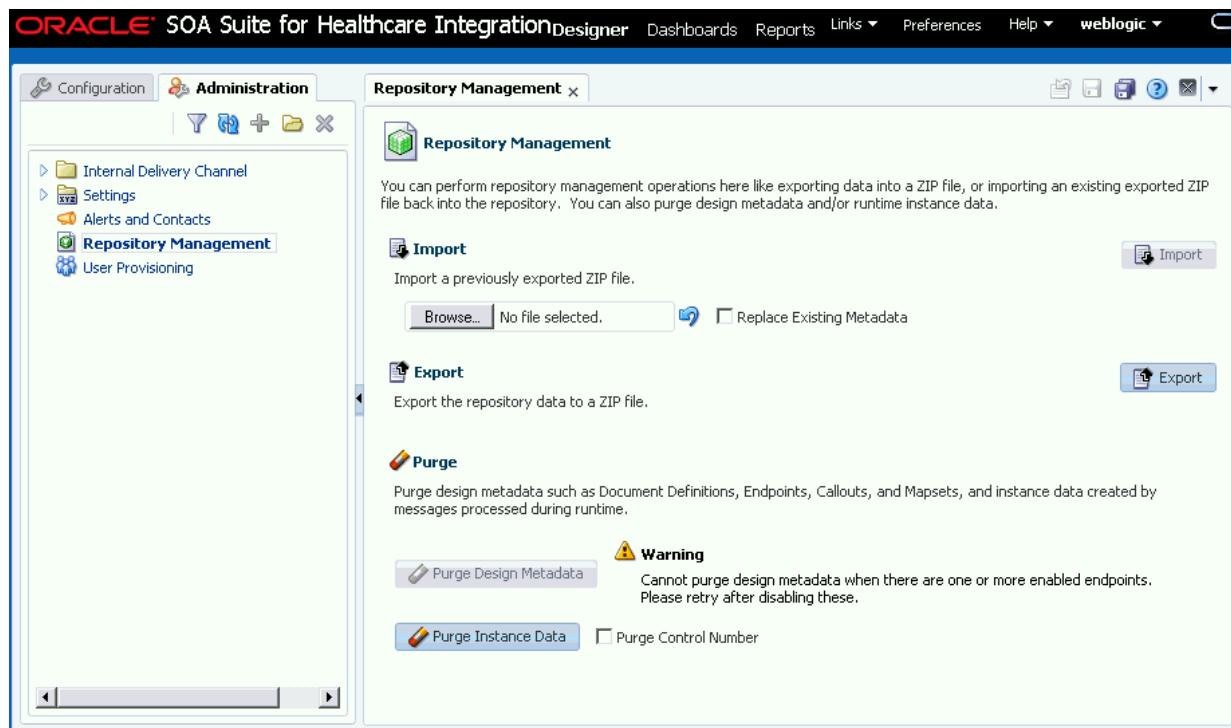
When you import metadata, the updates to your existing repository are incremental unless you select the **Replace Existing Metadata** option. To delete all existing data before importing metadata, click the **Purge Design Metadata** button.

> **Caution:**
>
> Complete export operations without interruption or idle time. Leaving the browser idle for more than a few minutes during export operations can cause file corruption.

To import repository metadata

1. Make sure the metadata repository ZIP file you want to import is available to your local system.

2. On the healthcare integration user interface, click the **Designer** tab and then click the **Administration** tab.

3. Click **Repository Management**.

4. In the Import section, click **Browse** to find the metadata repository ZIP file.

    If you are importing a ZIP file that contains multiple ZIP files within it, you must unzip the containing file and import each ZIP file separately.

5. To overwrite the current metadata in the Metadata Service (MDS) repository, select **Replace Existing Metadata**

    If this option is not selected, only new data is copied to the MDS repository.

6. Click **Import**.

    Depending on the size of the design-time repository contents, this process might take some time.

To export repository metadata

> **Note:**
>
> Do *not* manually edit exported files.

1. On the healthcare integration user interface, click the **Designer** tab and then click the **Administration** tab.

2. Click **Repository Management**.

3. Click **Export**.

    A dialog appears giving you the status of the export file generation.

4. Click **Continue**.

5. On the dialog that appears, select **Save File** and then click **OK**.

    The default file name is `MDS_EXPORT_DD_MM_YYYY.zip`.

6. Specify a name for the export file or accept the default name, browse to and select the a folder for the file, and then click **Save**.

> **Note:**
>
> Design time import and export can also be done using a command-line tool.

12.3 Purging Repository Data

Use the purge function to manage disk space and improve performance, and to remove test data from the repository.

To purge repository data

1. On the healthcare integration user interface, click the **Designer** tab and then click the **Administration** tab.

2. Click **Repository Management**.

3. Do one of the following:

   - To purge design-time metadata, such as endpoints, internal delivery channels, mapsets, callouts, and document definitions, click **Purge Design Metadata**.

     Note:

     This button is disabled if there are active endpoints in the healthcare integration repository. In this case, you must disable any active endpoints and return to this page to purge design-time data.

   - To purge all runtime instance data (that is, all messages), click **Purge Instance Data**. If you want to remove control numbers with this data, select **Purge Control Number** before clicking **Purge Instance Data**.

4. Click **OK** on the confirmation dialog.

   Note:

   Purging repository data can also be done using a command-line tool.

13

Configuring System Settings

This chapter describes how to configure the appearance of the healthcare integration user interface and how to configure runtime processing for healthcare integration applications.

This chapter includes the following topics:

- Configuring the Runtime Settings
- Configuring the User Interface Settings

13.1 Configuring the Runtime Settings

Runtime settings control how Oracle SOA Suite for healthcare integration handles certain functions, like sequencing, functional acknowledgments, batch processing, Oracle BAM enablement, default queues, and so on.

Runtime settings are shared between Oracle SOA Suite for healthcare integration and Oracle B2B; changing the settings for one also changes the settings for the other. Oracle B2B settings in Oracle Enterprise Manager also apply to Oracle SOA Suite for healthcare integration. For more information about Oracle B2B settings in Oracle Enterprise Manager, see "Configuring Oracle B2B" in *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.

Figure 13-1   Runtime Settings
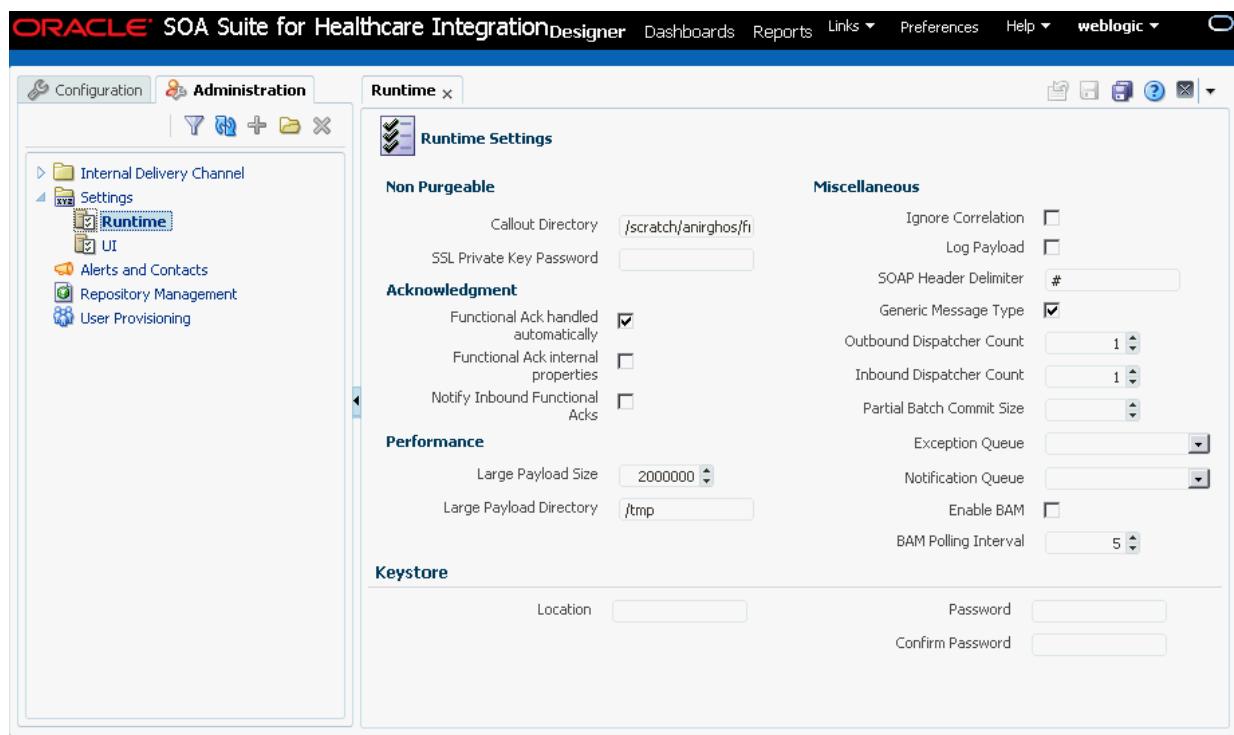
To configure the runtime settings

1. On the Oracle SOA Suite for healthcare integration user interface, click the **Designer** tab and then click the **Administration** tab.

2. Expand **Settings** and then select **Runtime**.

3. Modify the value of any of the properties listed in Table 13-1.

4. Do one of the following:

    *   To save your changes, click **Apply**.

    *   To change the settings back to their previous values, click **Revert**.

Table 13-1   Runtime Configuration Settings

Field	Description
Acknowledgment	

Table 13-1   (Cont.) Runtime Configuration Settings

Field	Description
Functional Ack Handled Automatically	An indicator of whether Oracle SOA Suite for healthcare integration automatically generates the functional acknowledgment (FA) for inbound HL7 messages. Select this option to automatically generate the FA; inbound FA messages are consumed when this option is selected.
	Clear this property if you do not want to automatically generate the FA document. The back-end application (middleware) must generate the FA and provide it to Oracle SOA Suite for healthcare integration as an outbound message. When this property is deselected, inbound FA documents are passed back to the back-end application.
	If the document does not require an FA (as indicated by the endpoint-level setting), then this property is ignored. This property is selected by default.
	If the FA is marked as expected in an endpoint, then the message is placed into the MSG_WAIT_FA state and the back-end application is expected to generate the FA and push it to Oracle SOA Suite for healthcare integration as an outbound message back to the partner.
	When **Functional Ack Handled Automatically** is not selected, **Notify Inbound Functional Acks** must also be deselected in order for the inbound FA to be sent to the back-end application. If **Notify Inbound Functional Acks** is selected and **Functional Ack Handled Automatically** is deselected, the incoming functional acknowledgment generates only a notification and the FA document itself is *not* sent back to the back-end application.
	The following limitations apply when generating the FA from the back-end application:
	• The FA is correlated with the original message based on the ReferToMsgID value set in the enqueue properties. The FA is correlated based on control numbers also.
	• If the FA indicates that there was an error in the received message, the status of the correlated message is not updated to indicate an error. The correlated message is updated to MSG_COMPLETE.
	These limitations are not present when the FA is generated by Oracle SOA Suite for healthcare integration (that is, when this property is selected).
Functional Ack Internal Properties	An indicator of whether to generate the internal properties structure in the functional acknowledgment using the original message internal properties.
	By default, this property is deselected, which means that the functional acknowledgment uses the original message-internal properties.

Table 13-1   (Cont.) Runtime Configuration Settings

Field	Description
Notify Inbound Functional Acks	An indicator of whether Oracle SOA Suite for healthcare integration sends an acknowledgment notification to the application when a functional acknowledgment is received. Select this property to send the acknowledgment notification.
	When **Functional Ack Handled Automatically** is not selected, **Notify Inbound Functional Acks** must also not be selected in order for the inbound FA to be sent to the back-end application. If **Notify Inbound Functional Acks** is selected and **Functional Ack Handled Automatically** is not selected, the incoming functional acknowledgment generates only a notification and the FA document itself is *not* sent back to the back-end application.
Miscellaneous	
Log Payload	An indicator of whether to log the payload in a diagnostic log. When this property is selected, the payload is logged, but this also depends on the log level setting. Error messages are logged by default. Payload logging is useful for diagnostic purposes, but might be undesirable for security reasons. By default, this property is not selected.
	Note: You must set the log-level setting in the Enterprise Management console to TRACE to be able to log payloads. The available log-level settings are TRACE, NOTIFICATION, INCIDENT_ERROR, ERROR, and WARNING.
Exception Queue	A JMS internal delivery channel for the host to use as the exception queue. Exception notifications are sent to the queue name you specify here. A null value for this parameter means that exceptions are sent to the default JMS queue (B2B_IN_QUEUE).
Notification Queue	A JMS internal delivery channel for the host to use as the notification queue. System level notifications (at the endpoint level) are sent to this queue. A null value for this parameter means that exceptions are sent to the default JMS queue (B2B_IN_QUEUE).
	The events that are notified:
	• Outbound Case: When the skip message limit is reached, endpoint is paused
	• Inbound Case: When there is a parse failure or a message validation error, endpoint is paused or connection is reset,
	To enable the notification of the system events, a new Server property must be added in the Oracle Fusion Middleware Enterprise Manager Control console. The property, hc.mllp.EnableEventNotification, takes the values true or false to enable and disable notification respectively. If the value is not set, by default the notification for system events is not enabled.
Generic Message Type	An indicator of whether Oracle SOA Suite for healthcare integration need to ignore the HL7 trigger event. By default, this property is selected and set to true.

Table 13-1   (Cont.) Runtime Configuration Settings

Field	Description
Outbound Dispatcher Count	The number of dispatchers used for handling the outbound messages. This is used in message sequencing.
Inbound Dispatcher Count	The number of dispatchers used for handling the inbound messages. This is used in message sequencing.
Partial Batch Commit Size	The number of records to be committed when there is a large number of business messages for a message exchange.
Enable BAM	An indicator of whether to send runtime information to Oracle BAM. For more information, see "Monitoring Instance Message Data With Oracle BAM" in the *Oracle Fusion Middleware User's Guide for Oracle B2B*.
BAM Polling Interval	The polling interval in minutes for Oracle BAM. This is ignored if Oracle BAM is not enabled.
Ignore Correlation	An indicator of whether to ignore correlation errors when an acknowledgment is received and correlation fails. Acknowledgments are correlated to the actual business message of the sender. If the correlation fails, an exception is generated and the acknowledgment processing stops. To ignore the correlation in this case and continue processing the acknowledgment, select this property. By default, this property is not selected.
Non Purgeable	
Note: The nonpurgeable parameters retain their values even after metadata repository is purged.	
Callout Directory	The directory for the callout JAR file location (if you do not use the default callout). The callout directory path cannot end with a forward or backward slash (/ or \). The default file location, /MyCalloutDir, is retained after purging the metadata repository.
SSL Private Key Password	The private key password to decrypt the public key of the private key - public key pair in case of secured data exchange by using SSL/TLS. You can leave this field blank if the password is same as the Keystore Password.
Performance	
Large Payload Size	The large payload size, in bytes. The default value is 2,000,000 (2MG).
Large Payload Directory	The directory to store large payloads. The default directory is /tmp. For Windows-based systems, change the directory to an appropriate directory, such as C:\temp.
Keystore	

Table 13-1   (Cont.) Runtime Configuration Settings

Field	Description
Location	The path where the keystore for managing trusted certificates and private-public key pairs are stored. For a two-way authentication, you require to configure two keystores, one at the client side to store server certificates and the keys, and the other at the server side to store client certificates and keys.
Password and Confirm Password	The password to access the keystore.

> **Note:**
>
> In release 12c, the auto-stack handler feature has been removed from the Oracle SOA Suite for healthcare integration user interface. Going forward, use the channel-level retry functionality to retry messages. Use the sequencing dashboard manageability functions such as "discarding" and "reprocessing" instead of the stack handler functions.

13.2 Configuring the User Interface Settings

User interface settings control the appearance of the reports and dashboards on the healthcare integration user interface.

Figure 13-2   User Interface Settings
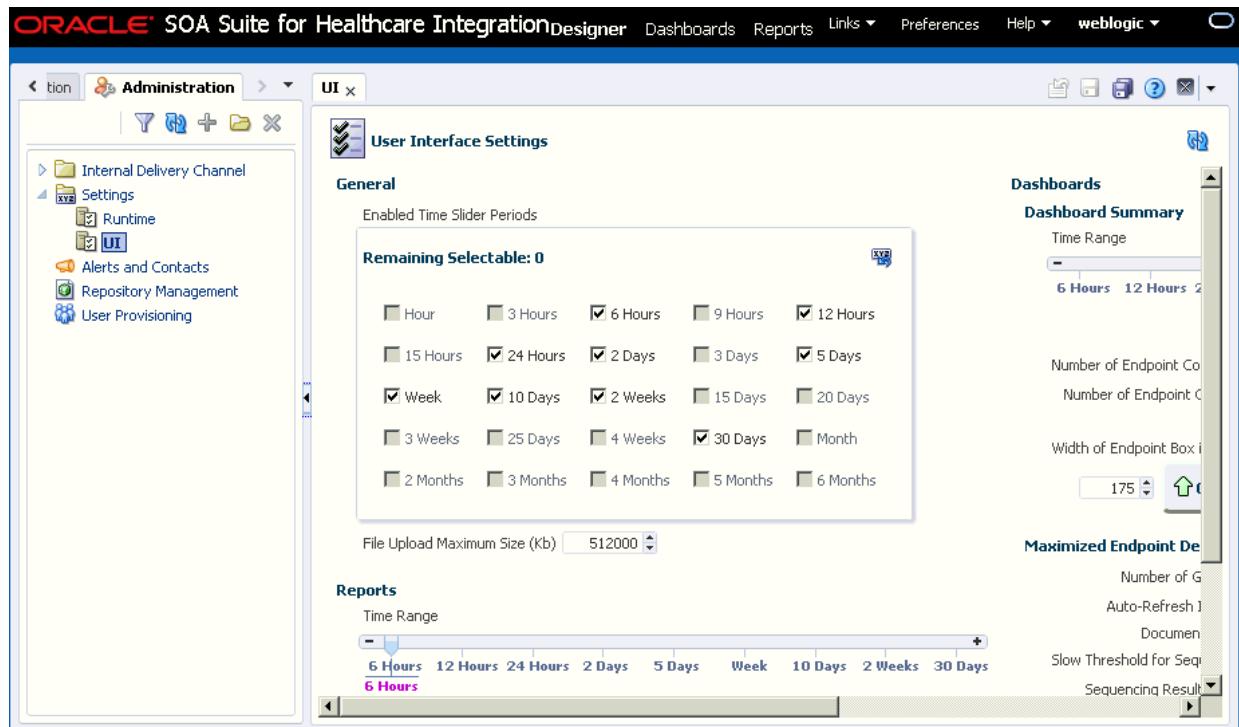
To configure the time slider periods for reports and dashboards

This procedure changes the values listed on the time sliders that are used on the Reports and Dashboard pages. You can select up to nine different time periods for the sliders.

Figure 13-3   Time Slider in the Reports Filter Panel

1. On the Oracle SOA Suite for healthcare integration user interface, click the **Designer** tab and then click the **Administration** tab.

2. Expand **Settings** and then select **UI**.

3. In the **Enabled Time Slider Periods** section, select the time periods you want the sliders to display.

    You can select up to nine periods, so be sure to clear time periods you do not require in order to make room for the ones you want to display. As you change your selections, you can see the changes to the dashboard summary slider on the right.

4. To restore the time period selections to the default, click **Restore Defaults**.

5. When you are done making changes, click **Apply** to save your changes.

To configure reports

You can configure certain display attributes for the Reports page, such as how many records to display, how often to auto-refresh, and payload display.

1. On the Oracle SOA Suite for healthcare integration user interface, click the **Designer** tab and then click the **Administration** tab.

2. Expand **Settings** and then select **UI**.

3. Change the value of any of the properties described in Table 13-2.

Table 13-2   Configuration Properties for Reports

Property	Description
Auto-Refresh Interval (secs)	The length of time (in seconds) the browser should wait between automatically refreshing information in a report.
Page Size	The number of messages to display on each page of a report.
Show Payload	An indicator of whether to display the message payload regardless of user permissions. Select **Yes** to display the payload; otherwise select **No**.
Payload Display Size (bytes)	The maximum number of bytes of each message payload to display. If a payload is larger than the number of bytes specified, none of the payload is shown.

4. When you are done making changes, click **Apply** to save your changes.

To configure the default time period for the dashboard summary

This procedure changes the default time period for the information shown on the Dashboard Summary pages. For example, if you select **5 Days**, when you open a dashboard you see a summary of the past five days for the selected endpoints; if you select **Week**, you see a summary of the past week.

1. On the Oracle SOA Suite for healthcare integration user interface, click the **Designer** tab and then click the **Administration** tab.

2. Expand **Settings** and then select **UI**.

3. In the Dashboard Summary section, move the pointer on the slider to the time period you want to use as the default for the Dashboard Summary pages.

4. When you are done making changes, click **Apply** to save your changes.

To configure endpoint details on the dashboard

You can configure the appearance of the Endpoint Details page of a dashboard. This is the page that appears when you click the Endpoint Details button for an endpoint on the Dashboard Summary page.

1. On the Oracle SOA Suite for healthcare integration user interface, click the **Designer** tab and then click the **Administration** tab.

2. Expand **Settings** and then select **UI**.

3. Change the value of any of the properties described in Table 13-3.

Table 13-3   Configuration Properties for Endpoint Details

Property	Description
Number of Gauge Columns	The number of message type gauges to display in each row of the Document Type Processed section.
Auto-Refresh Interval (secs)	The length of time (in seconds) the browser should wait between automatically refreshing endpoint details in a dashboard.
Document Type Display	An indicator of whether to display document type information in gauge format or in table format. Examples of each format are shown in Figure 13-4 and Figure 13-5.
Slow Threshold for Sequencing (secs)	The number of seconds that a sequence message can wait in the queue before processing is considered slow. After processing is considered slow, the status appears as yellow on the Endpoint Details page of a dashboard.

Figure 13-4   Document Type Information in Gauge Format

Figure 13-5   Document Type Information in Table Format

4. When you are done making changes, click **Apply** to save your changes.

14

Provisioning Users

This chapter discusses how to provision users by providing them roles using the Oracle Healthcare console.

The chapter contains the following topics:

- Creating Users

- Adding Users

- Editing, Viewing, and Deleting User Provisioning

- Provisioning Users for Payload Viewing

- Reverting User Provisioning Changes

14.1 Creating Users

The Oracle Healthcare administrator (the default login user name-password combination) can add additional users (registered users in the Weblogic default store) by using the Oracle Healthcare console. These users can log in to Oracle Healthcare and only view endpoint.

The following roles are available:

- Administrator role—Provides access to all Oracle Healthcare functionality

- Monitor role—Provides read access to reporting and metrics functionality only (use of the **Reports** link)

Users with the administrator role can access all Oracle Healthcare functions. Users with the monitor role have only read access to reports.

You can create users by using the Identity Store (in the Oracle Weblogic Server console), and then you can provision those users with specific roles in the console.

See "Task 1 Create a New User in the Identity Store" in the "Adding Trading Partner Users" in *Oracle Fusion Middleware User's Guide for Oracle B2B* for more information on creating users.

14.2 Adding Users

After you have created the registered users in the Oracle Weblogic Server console, you can add users to Oracle Healthcare by using the **User Provisioning** link under **Administration** in the **Designer** tab.

> **Note:**
>
> Due to security concerns, only registered users belonging to the WebLogic Administrators group are allowed to access the User Provisioning functionality in the Oracle Healthcare console. The **User Provisioning** link is not available for other users.

To add and provision a user:

1. Log on to the Oracle SOA Suite for healthcare integration console.

2. In the **Designer** tab, click **Administration** and then double-click **User Provisioning** on the left-hand panel.

3. In the User Provisioning pane on the right, click Add New User Provisioning for Healthcare UI (**+** button) under the Users table. This adds a blank row in the table.

4. Click the new row to select it.

5. Clear the existing prompt in the **Name** field and to select or provision a registered user, do one of the following:

   - Type the first few characters (case-insensitive by default) of the registered user name, and either click the **Browse** button (the magnifying glass icon) or press the Tab key.

     If there is only one match, it is automatically selected and the user name is populated in the Name field. If there are zero or more than one match, then the **Choose Registered User to Provision for Access to Healthcare UI** dialog box is displayed where you can refine the search, select one of the results, or both. Figure 14-1 displays the **Choose Registered User to Provision for Access to Healthcare UI** dialog box.

     Figure 14-1   Selecting Users from a List of All Matching Users

   - In case you are not sure about the user name that you want to select, click **Browse** or press Tab to display the **Choose Registered User to Provision for Access to Healthcare UI** dialog box that lists all the users available for provisioning as displayed in Figure 14-2.

Figure 14-2   Displaying a List of All Registered Users
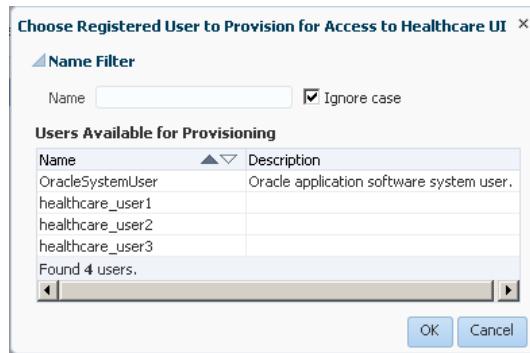
If the required user name is not listed in the dialog box, start typing the first few characters of the registered user name in the **Name** field of the dialog box, and the results table is automatically populated accordingly. Select the **Ignore Case** check box if you must perform a case-insensitive search. You can also clear the input field and press Tab to browse all available users for provisioning.

> **Note:**
>
> After you have selected a registered user, the Description field is automatically populated with the description that you have provided for the user in the WebLogic Administration Console. You cannot modify the description in the Oracle Healthcare console.

> **Note:**
>
> If you try to search for a user that is a part of the Oracle Weblogic Administrator group in the **Choose Registered User to Provision for Access to Healthcare UI** dialog box, an error message is displayed stating that users belonging to the Weblogic Administrators group need not be provisioned, as displayed in the following graphic.
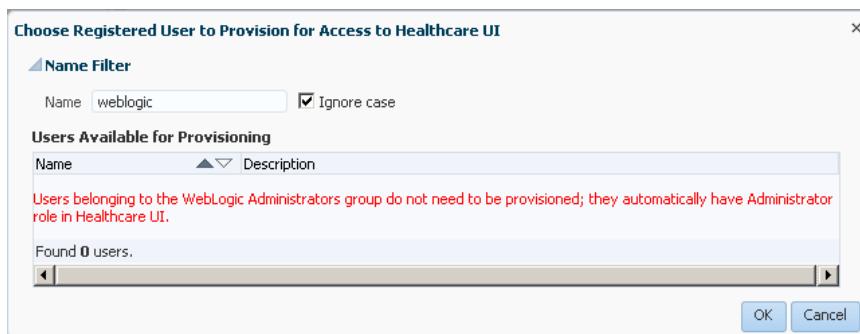
6. Select the role that you want to grant to the user from the **Role** list. The available values are **Administrator** and **Monitor** as shown in Figure 14-3.

Figure 14-3   Selecting the Role for the User
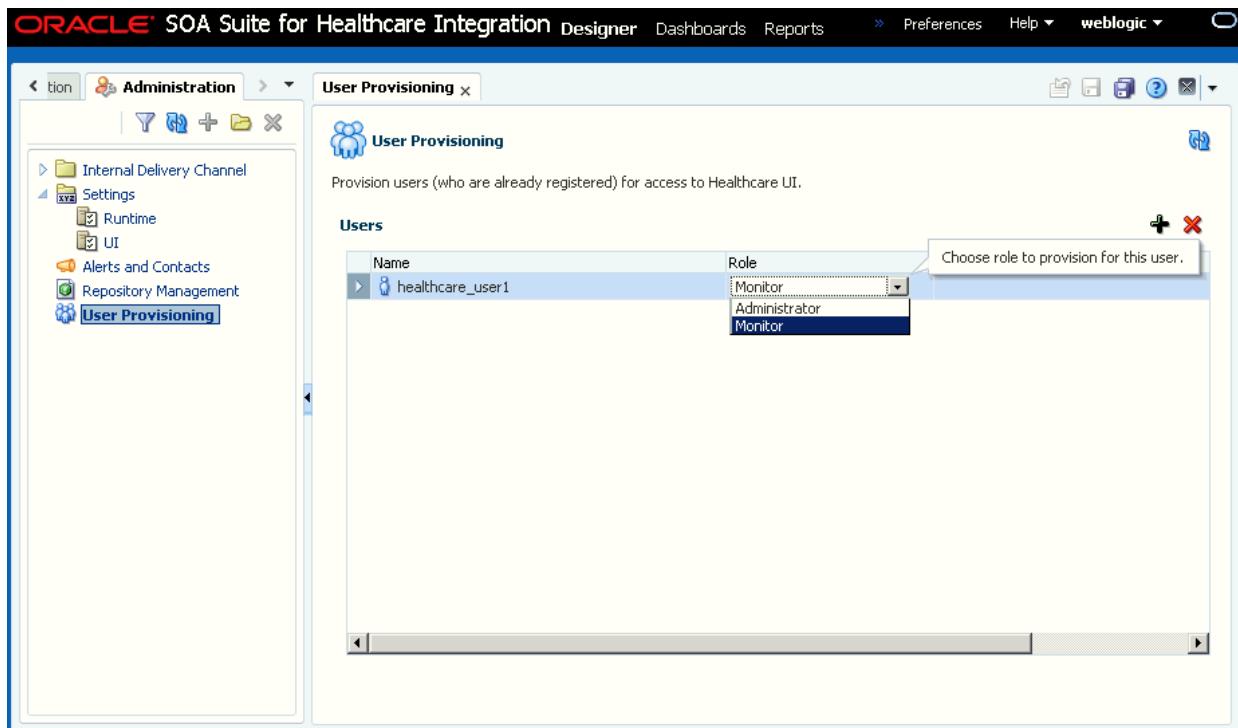
By default, all new users added are initially provisioned for the Monitor role in the Oracle Healthcare console, unless the user belongs to the WebLogic Administrators group, in which case the user is not available for provisioning (already added to the Administrators group).

Based on the role selected for the user, the user icon changes as shown in Figure 14-4.

Figure 14-4   Changing User Icon
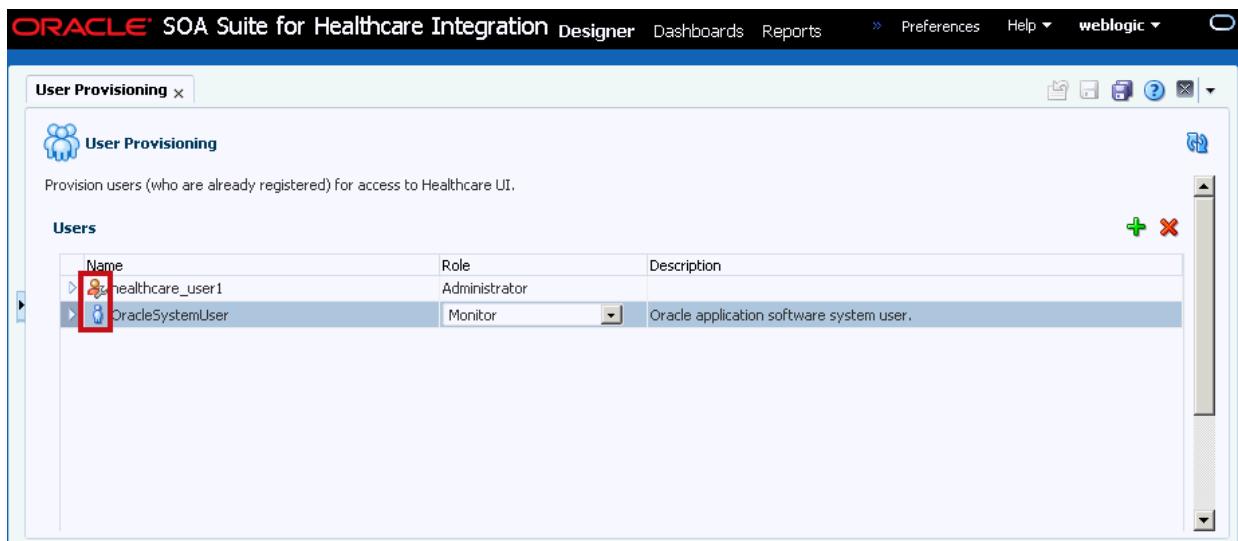
7. Click **Apply** to save the changes and complete the user provisioning.

Note:

After you have provisioned all the registered users for the Oracle Healthcare console, if you click Add New User Provisioning for Healthcare UI (**+** icon), the following window stating that no more users are available for provisioning is displayed.

Adding New User Provisioning...   ×

No more registered users are available for provisioning.

You can try applying or reverting any changes made and then click **Refresh** for an updated list of registered users.

OK

14.3 Editing, Viewing, and Deleting User Provisioning

You can edit, delete, and view user provisioning by using the Oracle Healthcare console.

To edit an existing user provisioning:

1. Log on to the Oracle SOA Suite for healthcare integration console.

2. In the **Designer** tab, click **Administration** and then **User Provisioning** on the left-hand panel.

3. Click the required user row in the Users table. This enables the **Role** list for editing.

4. Select the required role from the **Role** list.

To view user details:

1. Log on to the Oracle SOA Suite for healthcare integration console.

2. In the **Designer** tab, click **Administration** and then **User Provisioning** on the left-hand panel.

3. Click the Expand (down arrowhead) button to the left of the required user in the Users table to display the user details as shown in Figure 14-5.

Figure 14-5   Viewing User Details
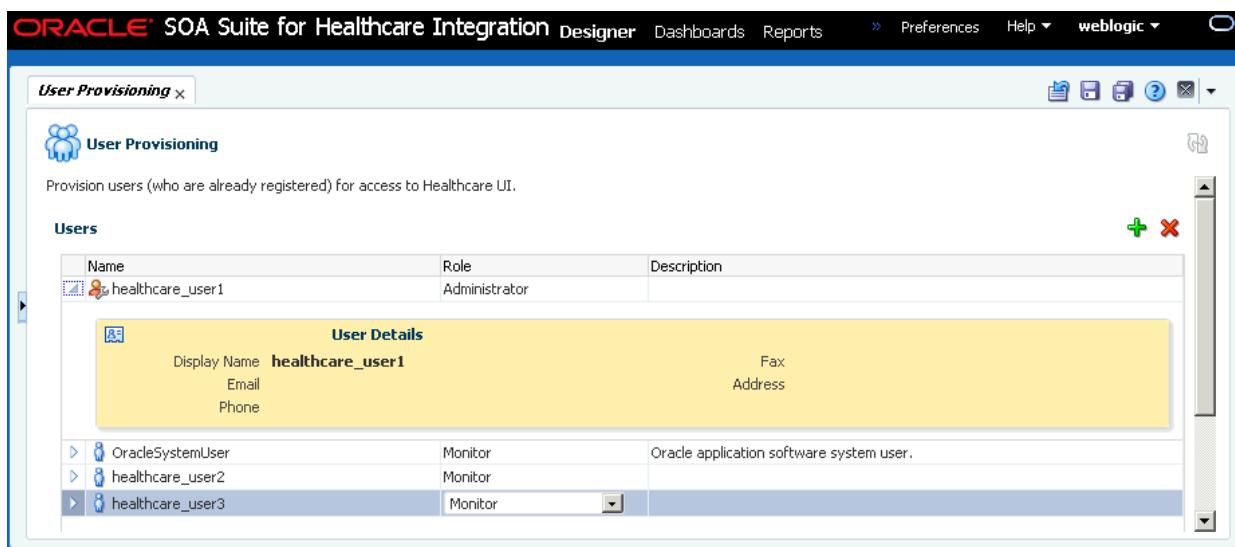
Note:

The user details are fetched from the Oracle WebLogic Security Realm that you have used when creating the user.

To delete an existing user provisioning:

1. Log on to the Oracle SOA Suite for healthcare integration console.

2. In the **Designer** tab, click **Administration** and then **User Provisioning** on the left-hand panel.

3. Click the corresponding row for the required user name in the Users table and click the Delete Selected User Provisioning (X) button. No confirmation is required, because you can easily revert the deletion.

Note:

Deleting a user provisioning deletes the provisioning for the registered user's access to Oracle Healthcare console only; it does not delete the user from the default Oracle WebLogic Security Realm.

14.4 Provisioning Users for Payload Viewing

Oracle Healthcare user provisioning for payload viewing adheres to several principles.

- Users who belong to WLS Administrators group can always view any document type payload and do not require any provisioning in the Oracle Healthcare console.

- Users who have been provisioned for Oracle Healthcare Administrator or Monitor role with *no specific* document type provisioning, can view *all* document type payloads.

- Users who have the Oracle Healthcare Administrator or Monitor role with *specific* document type provisioning can only view those particular document type payloads.

14.5 Reverting User Provisioning Changes

You can revert changes made to the user provisioning in Oracle Healthcare console by clicking the **Revert** button at the top right of the User Provisioning pane.

15

Enabling Web-Service-Based Message Exchange in Oracle Healthcare

This chapter describes how to enable Web-service-based message (typically SOAP-based) exchange between endpoints in Oracle SOA Suite for healthcare integration.

The chapter contains the following sections:

- Introduction to Web-Service-Based Message Exchange
- Exchanging SOAP-Based Service Messages with Custom WSDL
- Sending Custom SOAP Headers
- Sample Request-Reply Scenarios

15.1 Introduction to Web-Service-Based Message Exchange

Oracle Healthcare allows you to exchange Web service (SOAP) based messages between endpoints.You can exchange messages in both inbound and outbound direction. However, currently, this support is limited to SOAP 1.1 messages over HTTP only.

The Web service feature not only enables endpoints to receive or send messages, but also it is layered as a protocol implementation and supports other general features such as reporting, tracking, and auditing.

Many enterprises are increasingly having a requirement to integrate their endpoint file transfer and/or message exchange using Web service (in addition to Healthcare-specific protocols.)

15.2 Exchanging SOAP-Based Service Messages with Custom WSDL

The support for SOAP-based messages is available for both inbound and outbound directions.

You must create or upload a Web Service Definition Language (WSDL) file that you can customize according to your requirement.

15.2.1 Exchanging Outbound SOAP-Based Messages

To enable exchange of outbound SOAP-based messages, you must perform the following tasks:

- Uploading the WSDL
- Creating a document
- Creating an endpoint

- Attaching security policies

15.2.1.1 Uploading the WSDL

As the first task, you must upload the WSDL file that is required to register a Web service to exchange messages. You can upload the WSDL in the either of the following ways:

- Inline WSDL - a normal WSDL file where the XSD information is defined within the WSDL itself

- A ZIP file containing a WSDL file and an XSD file

- A ZIP file containing multiple WSDL and XSD files

To upload a WSDL:

1. Log on to the Oracle SOA Suite for healthcare integration console (http:// *<hostname>*:*<port>*/healthcare), where *hostname* is the name of the computer where Oracle SOA Suite is hosted and *port* is typically `8001` (in the case of a non-SSL connection).

2. Click the **Designer** tab and then click the **Configuration** tab.

3. Click to select the **WSDL** folder.

4. Click the **+** button (Create). The Create WSDL dialog box appears.

5. Specify a name for the WSDL such as TransmitWSDL. Ensure that the name does not contain any spaces or special characters. In the case of a Zipped WSDL, select the Zip file and then select the root WSDL.

6. For the Definition File field, click the **Browse** button to select the WSDL to be uploaded. For this example, the WSDL selected is TransmitDoc2way.wsdl. Click **OK** to complete the process.

   Figure 15-1 displays the newly created WSDL.

Figure 15-1   Uploaded WSDL
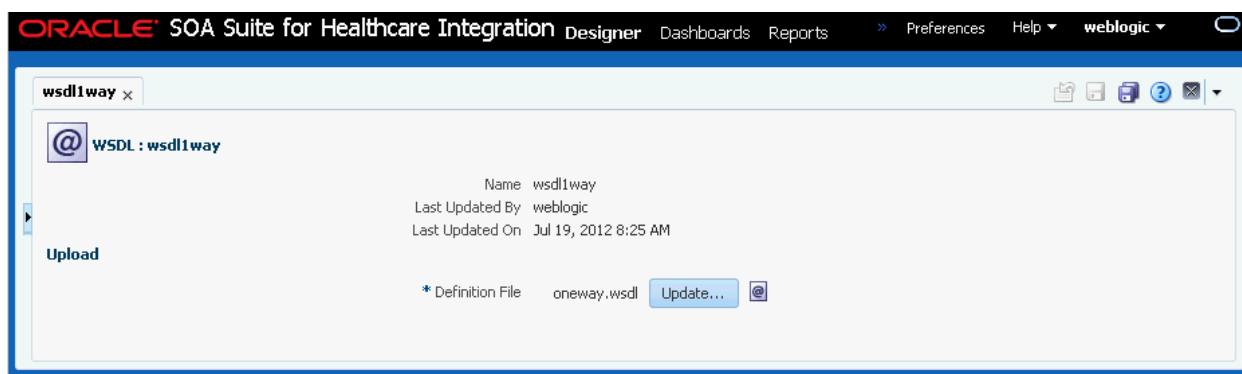
7. Click **Save** and then click **OK** in the confirmation dialog.

15.2.1.2 Creating a document

The next task is to create a document for the outbound flow. Please note that currently, the only supported document type is custom.

To create a custom document:

1. Create document definitions as specified in Creating Document Definitions. Specify:

   - Document Version - 1.0

   - Document Type - TransmitDocumentRequest

   - Document Definition - TransmitDocumentDef

2. Select **Use WSDL**.

3. Select the relevant WSDL Artifact, which in this case is **TransmitWSDL**.

4. Select the required WSDL Message, which in this case is the **TransmitDocumentsRequestMessage** as shown in Figure 15-2.

Figure 15-2   Creating Documents
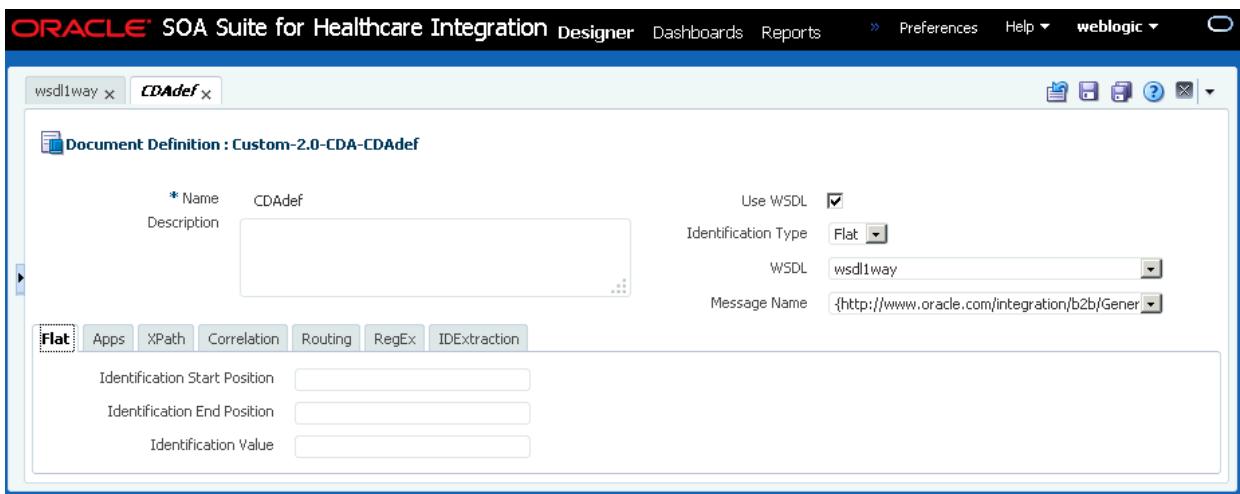
5. Click **Apply**.

15.2.1.3 Creating an endpoint

After you have created the custom document, you must create an endpoint and associate the endpoint with the custom document.

To create an endpoint:

1. In the Oracle Healthcare console, click the **Endpoint** folder under **Configuration** in the **Designer** tab.

2. Click the **+** button (Create). The Create Endpoint dialog box appears.

3. Select **WS-HTTP** from the Transport Protocol list.

4. Specify a name for the endpoint.

5. Select the **Synchronous** check box if you want to enable sending and receiving of documents by using the same connection. In this case, leave the check box as is.

6. Select **Outbound** from the Direction list.

7. Select the required WSDL from the WSDL list. In this case, select **TransmitWSDL**, which you uploaded in Uploading the WSDL.

> **Note:**
>
> If you do not want to upload and use a custom WSDL, you can select the **Use Generic SOAP** list. This should enable you to send any XML document over SOAP. Oracle SOA Suite for healthcare integration provides a pre-seeded generic WSDL that is used when you select **Use Generic SOAP**.

8. Select the available service (**TransmitsDocumentService** in this case)

9. Select the available port (**TransmitDocuments2WayPort**)

10. Select the SOAP action (available from the WSDL)

> **Note:**
>
> You can also provide string values for the preceding parameters manually. In the case of manually providing the service name, it should be in the following format, else there would be a validation error:
>
> {*namespace*}*ServiceName*

11. Enter the URL where the server is listening in the URL field.

    The endpoint configuration should look similar to Figure 15-3.

    Figure 15-3   Creating an Endpoint
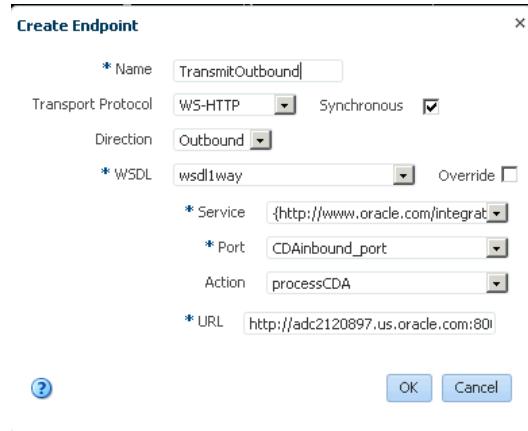
12. Click **OK**.

> **Note:**
>
> After the uploaded WSDL is used by an endpoint, you cannot update the WSDL file any more.

13. In the TransmitOutbound page, in the Document To Send section, click the + (Add) button to display the Document window.

14. Navigate to the required custom document definition (TransmitDocumentDef), select it, and click **OK** to associate the document with the endpoint. You can also drag-and-drop the document definition to the **Document To Send** box.

15. Click the **Transport Details** button to display the Transport Protocol parameters dialog box. Here, you can:

- Enter any additional HTTP headers, if required.

- Select **Omit WS Addressing Headers** if you want to discard WS Addressing headers as a part of the message to be exchanged.

- Select **Omit Oracle Default SOAP Headers** if you do not want Oracle Healthcare Web services outbound channel to send the default SOAP headers such as To, Doctype, and DocRevision as a part of the message as shown in Figure 15-4.

- In the **Advanced** tab, enter any additional SOAP headers, if required.

Figure 15-4   Transport Protocol Parameters
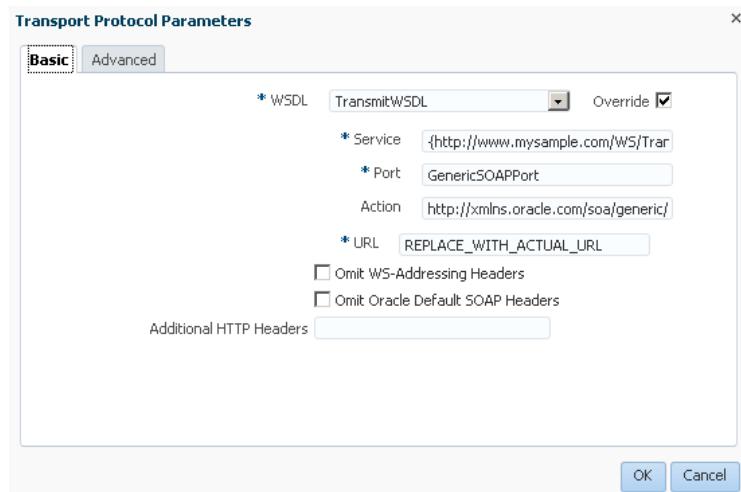
- Click **OK** to return to the endpoint page.

16. If the payload has to be validated against a WSDL or schema then enable the validation. Select the **Enabled** check box and click **Apply** to activate the endpoint as shown in Figure 15-5.

Figure 15-5   Configuring an Endpoint for Outbound Message Exchange
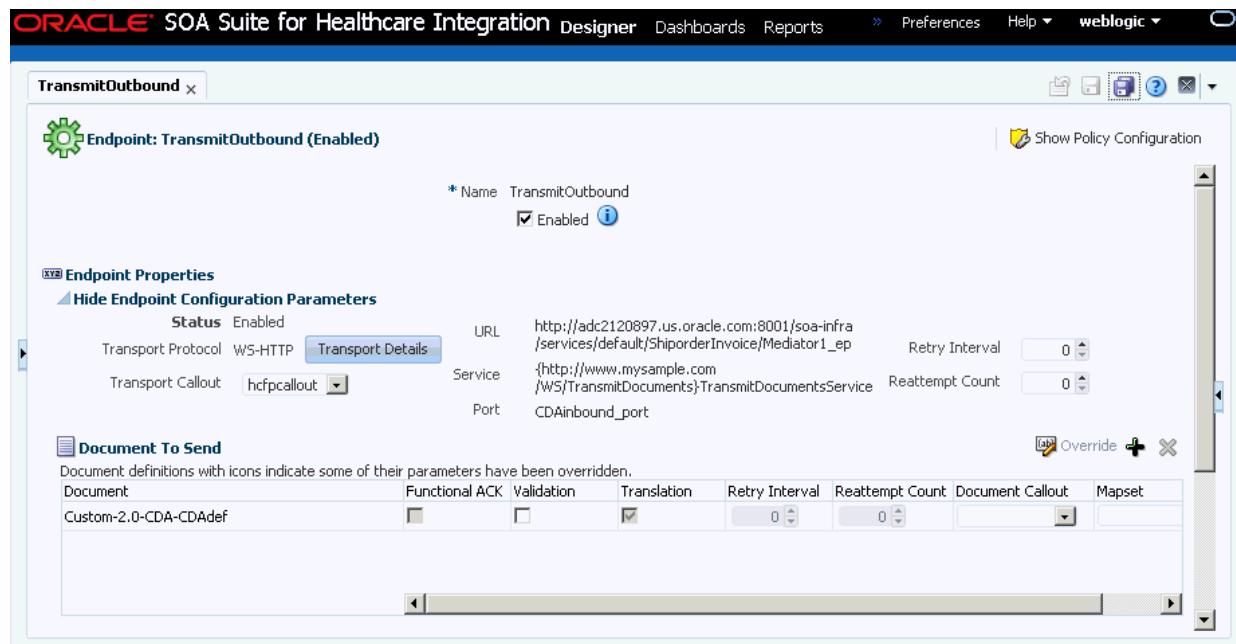
15.2.1.4 Attaching security policies

After you create an endpoint, you can attach security policies to it. Attaching security policies at the endpoint level rather than Global Policy Attachment (GPA) using the Oracle Enterprise Manager Fusion Middleware Control console enables you to maintain security policies at the Oracle Healthcare level locally.

By attaching policies locally, you can ensure that:

- Every endpoint has its own policy metadata

- Policies can be enabled or disabled along with endpoints

- Policies can be deleted along with endpoints

To attach security policies locally to an endpoint:

1. Open the endpoint that you created in Creating an endpoint.

2. On the endpoint page, click the **Show Policy Configuration** button on the top right-hand corner. The Policy Configuration section appears.

3. From the list of available Web services policies under Available Policies, select the policy that you want to attach to the endpoint. You can select multiple policies by pressing the Control key and clicking the policy names.

4. Click **Attach** to attach the selected policy or policies to the endpoint. You can click the **Attach All button** to attach all the available policies to the endpoint as shown in Figure 15-6.

Figure 15-6   Attaching Security Policies
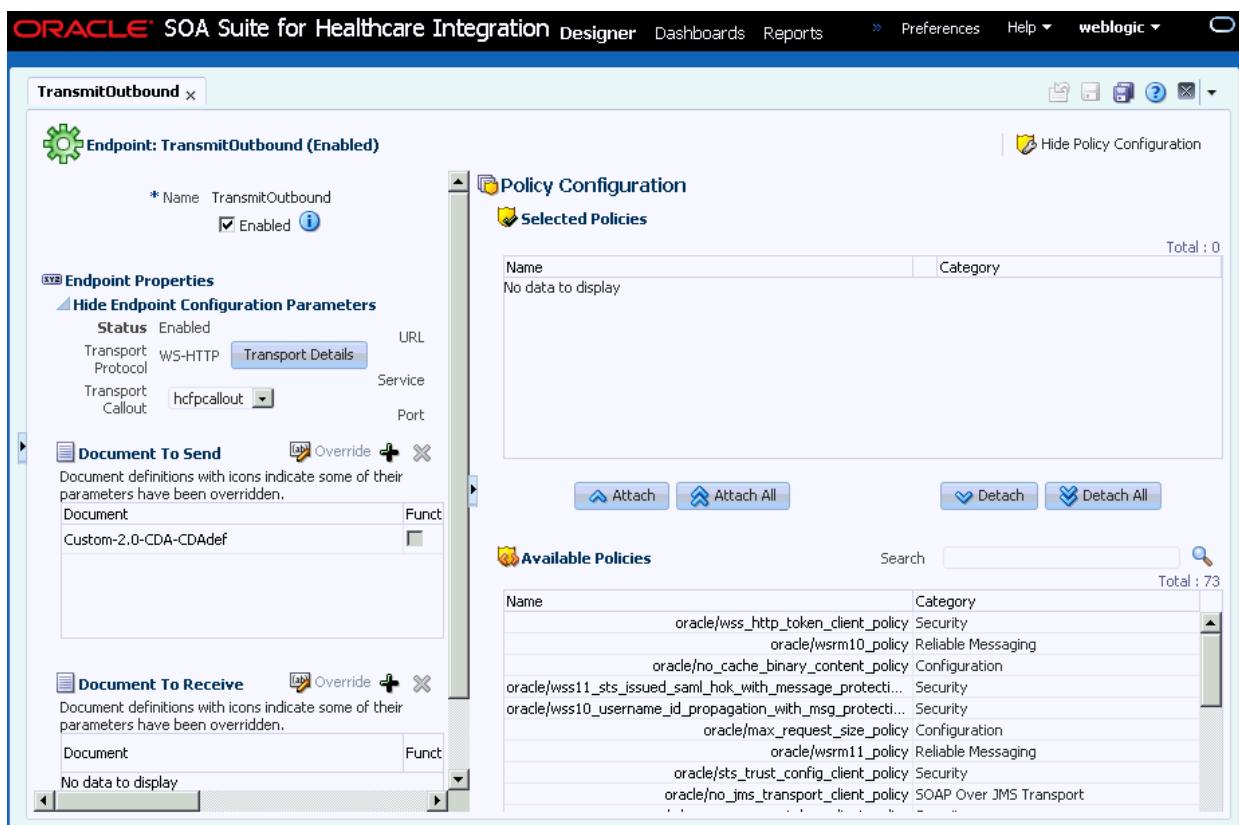
Note:

Some policies are contradictory to each other. If you try to attach such contradictory policies together to an endpoint, you get an error.

5. Click the **Save** button.

15.2.2 Exchanging Inbound SOAP-Based Messages

To exchange inbound SOAP-based messages, you must perform the following tasks:

- Uploading the WSDL

- Creating a document for the inbound flow

- Creating an endpoint for inbound message exchange

- Attaching security policies for inbound message exchange

15.2.2.1 Uploading the WSDL

This task is the same as Uploading the WSDL of the outbound case.

15.2.2.2 Creating a document for the inbound flow

This task is the same as Creating a document of the outbound case.

15.2.2.3 Creating an endpoint for inbound message exchange

After you have created the custom document, you must create an endpoint and associate the endpoint with the custom document.

This task is similar to Creating an endpoint of the outbound case with the following differences:

- In the Create Endpoint dialog box, specify `TransmitInbound` as the endpoint name and select **Inbound** as the direction.

- In the TransmitInbound page, add the custom document in the Document To Receive section.

15.2.2.4 Attaching security policies for inbound message exchange

After you have created an endpoint, you can attach security policies to it. This task is similar to Attaching security policies of the outbound case.

Note:

To ensure that the that the Web services have been registered for listening messages, access the following URL, log on by providing the user name and password, and check if the Web service is listed in the list of registered Web services:

http:*<host>*:*<port>*/b2b/services

You must create users (along with their passwords) in the Oracle Weblogic Server console. Access the following link to know more about creating users in the Oracle Weblogic Server console:

http://docs.oracle.com/cd/E23943_01/apirefs.1111/e13952/
taskhelp/security/ManageUsersAndGroups.html

The Web service is listed in the following format:

ws/*WebService*

You can download the WSDL from the specific service by clicking the respective WSDL link as shown in the following graphic.

15.3 Sending Custom SOAP Headers

Oracle Healthcare allows you to send custom SOAP headers as a part of your outbound Web-service-based messages.

In the case of an outbound JMS channel, the sender trading partner can send multi-level custom SOAP headers as the following sample:

```
<CustomSOAPHeader xmlns="http://schemas.xmlsoap.org/soap/envelope/">
   <hello xmlns="http://xmlns.oracle1.com/soa1/generic/soap">
       <name xmlns="http://MY_NAME_SPACE">
         <firstname>John</firstname>
         <lastname>Doe</lastname>
       </name>
```

```
      </hello>
</CustomSOAPHeader>
```

In the case of an outbound fabric, the sender can send multi-level custom headers as the following sample:

```
<bpelx:inputProperty name="hc.customSOAPHeaders" expression="'<CustomSOAPHeader
xmlns="http://schemas.xmlsoap.org/soap/envelope/">
   <hello xmlns="http://xmlns.oracle1.com/soa1/generic/soap">
      <name xmlns="http://MY_NAME_SPACE">
        <firstname>John</firstname>
        <lastname>Doe</lastname>
      </name>
   </hello>
</CustomSOAPHeader>
```

15.4 Sample Request-Reply Scenarios

This section discusses the synchronized request-reply scenarios when using a Web service or the Generic SOAP.

The section contains:

- Outbound Synchronization: Composite

- Inbound Synchronization: Composite

- Outbound Synchronization: JMS Queues

- Inbound Synchronization: JMS Queues

15.4.1 Outbound Synchronization: Composite

In the case of outbound synchronization with a composite:

- **Web Service:** Relies on the composite to decide whether the synchronization is one-way, or if the request/reply pattern is used. If soapAction is provided, it is used only to overwrite the HTTP soapAction headers.

15.4.2 Inbound Synchronization: Composite

In the case of inbound synchronization with a composite:

- **Web Service:** Uses soapAction from the HTTP header to decide whether it is a one-way or request/reply operation specified by the WSDL. In the case of a non-responsive payload, an error is reported for request/reply.

15.4.3 Outbound Synchronization: JMS Queues

In the case of outbound synchronization with a JMS queue:

- **Web Service:** In the case of custom WSDL, uses soapAction (provided from the back end or configured in the endpoint) to decide whether it is a one-way or request/reply operation specified by the WSDL. Reply is sent to the back end by using the internal delivery channel configuration. In the case of a generic SOAP WSDL, the transport parameters of the endpoint decides on the one-way or request/reply operation.

15.4.4 Inbound Synchronization: JMS Queues

- **Web Service:** Uses `soapAction` from the HTTP header to decide whether it is a one-way or request/reply operation specified by the WSDL. In the case of **Sync**, a reply must be generated in a Oracle SOA Suite for healthcare integration callout to be sent to the back end. In the case of a generic SOAP WSDL, the decision is based on the transport parameters specified in the endpoint.

16

Oracle Healthcare Command-Line Tools

This chapter describes the Oracle Healthcare command-line tools that are available for a number of tasks such as purging data, importing and exporting data, batching operations, updating and deleting endpoints, and so on.

This chapter contains the following topics:

- Prerequisites for Running the Command-line Tools
- Purging Data
- Importing Data
- Exporting Data
- Batching Operations
- Resubmitting a Message
- Scheduling Endpoint Downtime
- Updating Endpoints
- Pausing and Resuming Endpoints
- Deleting Endpoints
- Updating Keystore

16.1 Prerequisites for Running the Command-line Tools

Do the following before using the command-line tools:

1. Set the `ORACLE_HOME`, `ANT_HOME`, and `JAVA_HOME` environment variables.

   `ORACLE_HOME` is set to your Oracle SOA Suite install in the Oracle Fusion Middleware installation directory. For example if the default shell is CSH:

   ```
   setenv ORACLE_HOME <Oracle SOA install directory>
   setenv ANT_HOME $ORACLE_HOME/../modules/org.apache.ant_1.7.1
   set JAVA_HOME $ORACLE_HOME/<jdk install directory>
   ```

2. Create `jndi.properties`.

   ```
   cd $ORACLE_HOME/bin
   ant -f ant-hcfp-util.xml hcfpcreate-prop
   ```

3. Edit the `jndi.properties` file to include the `weblogic` password.

> **Note:**
>
> 1. Command-line tools are for administrator use only. No security or permission checks are performed to prevent the logged-in user from purging, importing, or exporting data.
>
> 2. After running any command-line tool, you should gain log on to the Oracle Healthcare console. The Oracle Healthcare console caches some metadata and any command-line action which might have updated the metadata could lead to invalid cached data. Therefore, it is advisable to always re-login into the Oracle Healthcare console after using command-line operations.
>
> 3. All of the command-line tools can be run without any JNDI credentials. To restrict the command-line tools from anonymous use, provide the following information in the `jndi.properties` file:
>
> ```
> java.naming.security.principal=weblogic
> java.naming.security.credentials=weblogic_password
> ```

> **Note:**
>
> For any Oracle Healthcare command-line utility, by default, the ANT run shows `BUILD SUCCESSFUL` and system code `0` (zero) is returned even in the case of client or server side error. In case you want the ANT run to FAIL for each client or server side error, then you must set `exitonerror` parameter to `true`.
>
> You can do this in either of the following two ways:
>
> - Specify `-Dexitonerror=true` on ANT command line
>
>   ```
>   ant -f ant-hcfp-util.xml b2bpurge -Dagreement=<AGR_NAME> -Dmode=DT -Dexitonerror=true
>   ```
>
> - Setting the `exitonerror` parameter as global setting:
>
>   Create a properties file called ant_general.properties with the value `exitonerror=true` in the directory where ant-hcfp-util.xml is present (`$ORACLE_HOME`/bin).
>
> With this configuration, for all client or server errors, all ANT commands fail with the message `BUILD FAILED` and the system exit code is set to `-1`.

16.2 Purging Data

This utility is used to purge design and runtime data from the Oracle Healthcare repository. This utility also provides the archiving feature by default. If Archive is set to `YES`, then the an initial setup is required.

To do an initial setup:

1. On the machine running the database, create a directory in which to dump the archive file. For example:

   ```
   mkdir /tmp/archive
   ```

2. Grant permissions to this directory so that the database process can write to it.

```
chmod 777 /tmp/archive
```

3. Log in to the database as `sysdba`.

```
sqlplus / as sysdba
```

4. Set up the `HCFP_EXPORT_DIR`.

```
SQL> create or replace DIRECTORY HCFP_EXPORT_DIR as '/tmp/archive'
```

5. If your SOA schema user is `hcfp_soainfra`, the user must be granted permission for the export.

```
SQL> grant read, write on directory HCFP_EXPORT_DIR to hcfp_soainfra;
SQL> grant exp_full_database  to hcfp_soainfra;
```

The following utility purges both design-time and runtime data and resets the environment to the installation time.

```
ant -f ant-hcfp-util.xml hcfppurge
```

Table 16-1 lists the options for this utility.

Table 16-1   Options for hcfppurge utility

Option	Description	Domain	Required
mode	Specifies purging design-time or runtime data (see Note below). Runtime options: `msgstate/start/end/purgecontrolnumber` Design-time options: `host`	DT RT	Yes
msgState	Deletes messages with the specified message state. Used for runtime data.	MSG_COMPLETE MSG_ERROR MSG_WAIT_TRANSMIT MSG_WAIT_FA MSG_WAIT_BATCH	No. If `msgstate` is present, then `start` and `end` must be used.
purgecontrolnumber	Deletes control numbers. Used for runtime data.	true false (default)	No
fromdate	Deletes all messages, which created on or after this date.	Date format dd/mm/yyyy hh:mm AM/PM	No
todate	Deletes all messages, which created on or before this date.	Date format dd/mm/yyyy hh:mm AM/PM	No
direction	Direction of the message		No
msgtype	Type of the message		No
archive	Should archive		Default value is true

Table 16-1   (Cont.) Options for hcfppurge utility

Option	Description	Domain	Required
archivename	File name of archived file		No

> **Note:**
>
> When only `-Dmode=RT -Dtp=endpoint_name` options are used, this option deletes all records matching endpoint name with SenderName or in Receiver Name.

Example - Removes Design-Time Data

```
ant -f ant-hcfp-util.xml hcfppurge -Dmode=DT
```

Example - **Purges runtime Data**

```
ant -f ant-hcfp-util.xml hcfppurge -Dmode=RT -Darchive=false
```

Example - **Purges runtime Data, Including Control Numbers**

```
ant -f ant-hcfp-util.xml hcfppurge -Dmode=RT -Dpurgecontrolnumber=true
```

Example - Purges Messages with the Specified State Between the Specified Dates

```
ant -f ant-hcfp-util.xml hcfppurge -Dmode=RT -Dfromdate="01/02/2009 12:00 AM" -
Dtodate="10/02/2009 12:00 AM" -Dmsgstate=MSG_COMPLETE -Darchive-false
```

> **Note:**
>
> When using `archivename` the value must be a unique file name. An existing file name used with `archivename` throws an exception.

16.3 Importing Data

The `hcfpimport` utility imports the Oracle Healthcare metadata ZIP file to the repository. Basic validation is performed, but it is not a complete validation as with deployment validation. No data is overwritten unless you use the `overwrite` option.

> **Note:**
>
> No security or permission checks are performed to prevent the logged-in user from importing data.

The following usage imports data from `tmp/export.zip` to a location on the same server without overwriting.

```
ant -f ant-hcfp-util.xml hcfpimport -Dlocalfile=true -Dexportfile="/tmp/export.zip"
```

Table 16-2 lists the options for this utility.

Table 16-2   Options for hcfpimport utility

Table 16-2 (Cont.) Options for hcfpimport utility

Option	Description	Domain	Required
exportfile	Location of the export (ZIP) file	-	Yes
overwrite	Overwrites the existing business elements. For example, an existing endpoint with the same endpoint name as an endpoint in the import file is replaced if this option is set to true.	true false (default)	No
localfile	If the export file location exists on the server, then set this option to true to improve performance. The export file must be on the server on which Oracle Healthcare is running.	true false (default)	No
active	Enable all the endpoints after import	true false (default)	No

16.4 Exporting Data

The hcfpexport utility exports metadata from the Oracle Healthcare repository. If no options are specified, then the entire repository is exported.

> **Note:**
>
> No security or permission checks are performed to prevent the logged-in user from exporting data.

The following usage exports the entire repository (without policy details) to /tmp/export.zip if no other options are specified.

```
ant -f ant-hcfp-util.xml hcfpexport
```

Table 16-3 lists the options for this utility.

Table 16-3 Options for hcfpexport utility

Option	Description	Domain	Required
exportfile	Location of the ZIP file where the exported data is stored	/tmp/export.zip (default)	No
endpoint	Name of the endpoint		No
policies	Set to true to export the entire repository with user and role details, which is required for the policy store. A warning is displayed to remind you to export the policy store also.	true false (default)	No

Table 16-3   (Cont.) Options for hcfpexport utility

Option	Description	Domain	Required
localfile	Set to `true` for improved performance *if* the export file is on the same computer as Oracle Healthcare.	`true` `false` (default)	No

Example - Exports entire repository with policy details to /tmp/export.zip

```
ant -f ant-hcfp-util.xml hcfpexport -Dexportfile="/tmp/export.zip" -Dpolicies=true
```

Example - **Exports entire repository w/o details to /tmp/exportinserver.zip on same server**

```
ant -f ant-hcfp-util.xml hcfpexport -Dexportfile="/tmp/exportinserver.zip" -
Dlocalfile=true
```

Example - **Exports the Endpoint Admission to /tmp/export.zip**

```
ant -f ant-hcfp-util.xml hcfpexport -Dexportfile="/tmp/export.zip" -
Dendpoint=Admission
```

16.5 Batching Operations

The command line `hcfpbatch` utility enables you to create and delete batches, based on various criteria. This is an ANT-based command, and provides the flexibility to selectively set the criteria to create batches.

The usage is as follows:

```
ant -f ant-hcfp-util.xml hcfpbatch -Dendpoint=<EndpointName> -
Dbatchtime=<batchTriggerTime> -Dbatchname=<batchName> -
Ddocument=<documentProtocolName> -Ddocrevision=<docRevision> -Ddoctype=<docType> -
Disrepetitive=<true|false>
```

Table 16-4 lists the options for this utility.

Table 16-4   Options for hcfpbatch utility

Option	Description	Domain	Required
endpoint	Endpoint name.	Specify the endpoint name.	Yes[1]
batchname	Batch name.	Specify a name for the batch.	Yes
batchtime	Batch trigger time.	The trigger time can be a cron String or the date in dd/MM/yyyy HH:mm AM/PM format.	Yes
document	Document Protocol name.	Values: EDI_EDIFACT, EDI_X12	Yes
docrevision	Document revision.	-	Yes
doctype	Document type.	-	Yes

Table 16-4   (Cont.) Options for hcfpbatch utility

Option	Description	Domain	Required
isrepetitive	To enable repetitive batching when batch is created using cron string.	Values: `true` or `false`	No
mode	Mode	Set to `deletebatch` to delete the batch.	No

[1] To create a batch the endpoint, batchtime, batchname, document, docrevision, doctype options are required, to delete a batch only the mode and batchname options are required.

> **Note:**
>
> While creating a cron job for the command-line operation, you must follow the cron syntax.

Example - Creates a batch operation

This command creates a batch operation with FileEndpoint for the X12/4010/850 document that is executed in a repetitive mode for the given cron String.

```
ant -f ant-hcfp-util.xml hcfpbatch -Dendpoint=FileEndpoint -Dbatchtime="0 4850 11 7
5 ? 2010" -Dbatchname=batch1234 -Ddocument=EDI_X12 -Ddocrevision=4010 -Ddoctype=850 -
Disrepetitive=true
```

Example - Creates batches for multiple document types

```
ant -f ant-hcfp-util.xml hcfpbatch -Dendpoint=FileEndpoint -Dbatchtime="0 58 11 7
5 ? 2010" -Dbatchname=batch1234 -Ddocument=EDI_X12 -Ddocrevision=4010 -
Ddoctype=850,997
ant -f ant-hcfp-util.xml hcfpbatch -Dendpoint=FileEndpoint -Dbatchtime="07/05/2010
11:45 AM" -Dbatchname=batch1234 -Ddocument=X12 -Ddocrevision=4010 -Ddoctype=850,997
```

Example - Deletes a batch operation

```
ant -f ant-hcfp-util.xml hcfpbatch -Dmode=deletebatch -Dbatchname=batch1234
```

Example - **Using special characters in batchtime option**

If the value for the batchtime contains special characters such as * or #, then the character must be escaped using double quotation marks.

```
ant -f ant-hcfp-util.xml hcfpbatch -Dendpoint=FileEndpoint -Dbatchtime='0
5,10,15,20,25,30,35,40,45,50,55,59 "*" "*" "*" ? 2010' -Dbatchname=batch1234 -
Ddocument=EDI_X12 -Ddocrevision=4010 -Ddoctype="850,855" -Disrepetitive=true
```

16.6 Resubmitting a Message

This utility resubmits an application message or a wire message for a selected business message.

```
ant -f ant-hcfp-util.xml hcfpresubmit
```

> **Note:**
>
> The resubmit functionality for payload rectification only works for Outbound Message for AppMessage only. This is not valid for inbound cases.
>
> ```
> ant -f ant-hcfp-util.xml hcfpresubmit -Dmsgsource=APPMSG -Dmsgid=12345 -
> Dpayloadpath=/scratch/<username>/fmwhome/AS11gR1SOA/bin/3a4_req.xml
> ```

Table 16-5 lists the options for this utility.

Table 16-5   Options for hcfpresubmit utility

Option	Description	Domain	Required
direction	The direction of the message	INBOUND OUTBOUND	No
msgsource	The message source	APPMSG (Default) WIREMSG	Yes
msgid	The message ID	Can be multiple message IDs separated by comma	
doctype	Document Type		
msgstate	Message State		
fromdate	The sendTimestamp of the message	Date format to be provided within Double Quotes dd-mm-yyyy hh:mm AM/PM **Note:** This cannot be a future date.	
todate	The sendTimestamp of the message	Date format to be provided within Double Quotes dd-mm-yyyy hh:mm AM/PM **Note:** todate should be greater than fromdate. You can provide both the dates.	

Table 16-5 (Cont.) Options for hcfpresubmit utility

Option	Description	Domain	Required
payloadpath	This option is applicable for outbound application message resubmission, by providing the rectified file path.		

Example - **Resubmits an Outbound Message with Message ID 12345**

```
ant -f ant-hcfp-util.xml hcfpresubmit -Ddirection=OUTBOUND -Dmsgsource=APPMSG -
Dmsgid=12345\
```

Other Examples

```
ant -f ant-hcfp-util.xml hcfpresubmit -Dmsgsource=APPMSG -Ddoctype=850
ant -f ant-hcfp-util.xml hcfpresubmit -Dmsgsource=APPMSG -Dfromdate="29/11/2009 5:40
AM" -Dtodate="30/11/2009 7:39 AM"
ant -f ant-hcfp-util.xml hcfpresubmit -Dmsgsource=WIREMSG -Dmsgstate=MSG_ERROR

ant -f ant-hcfp-util.xml hcfpresubmit -Dmsgsource=APPMSG -Dfromdate="29/11/2009 5:40
AM" -Dtodate="30/11/2009 7:39 AM" -Ddirection=OUTBOUND
ant -f ant-hcfp-util.xml hcfpresubmit -Dmsgsource=APPMSG -Dfromdate="29/11/2009 5:40
AM" -Dtodate="30/11/2009 7:39 AM" -Ddirection=INBOUND
ant -f ant-hcfp-util.xml hcfpresubmit -Dmsgsource=APPMSG -Dmsgid=12345 -
Dpayloadpath="/tmp/850.xml"
```

16.7 Scheduling Endpoint Downtime

Various Endpoint parties schedule their downtimes for different reasons and notify their partners about the downtime. During a downtime, parties sending the messages might not reach the destination. Scheduling the Endpoint downtime ensures that the messages are not delivered during that period, yet the messages are processed by Oracle Healthcare so that the messages are ready for delivery when the Endpoint party comes up after the downtime.

The following utility schedules downtime for an endpoint.

```
ant -f ant-hcfp-util.xml hcfpschedule
```

Table 16-6 lists the options for this utility.

Table 16-6 Options for hcfpschedule utility

Option	Description	Domain	Required
mode	Indicates if the script schedules or unschedules a downtime.	schedule (default) unschedule	Yes
schedulename	A descriptive name for the scheduled downtime	-	Yes
endpoint	Endpoint name	-	Yes (except in unschedule mode)

Table 16-6   (Cont.) Options for hcfpschedule utility

Option	Description	Domain	Required

`fromdate`	The date and time at which to begin the downtime.	Date format to be provided within Double Quotes
		
		`dd/mm/yyyy`
		`hh:mm AM/PM`
			No

`todate`	The date and time at which to end the downtime.	Date format to be provided within Double Quotes
		
		`dd/mm/yyyy`
		`hh:mm AM/PM`
			No

`extend`	Extends a previously created schedule.	`true`	No

The following are examples of scheduling endpoint downtime using the `hcfpschedule` utility. The command need not be in a single line.

Example - **Schedules endpoint downtime for all channels of GlobalChips from "14/05/2010 00:14 AM" to "14/05/2010 00:17 AM"**

```
ant -f ant-hcfp-util.xml hcfpschedule
-Dendpoint="GlobalChips"
-Dfromdate="14/05/2010 00:14 AM"
-Dtodate="14/05/2010 00:17 AM"
-Dschedulename= "Maintenance"
```

Example - Un-schedules the scheduled event

```
ant -f ant-hcfp-util.xml hcfpschedule
-Dmode=unschedule
-Dschedulename="Maintenance"
```

16.8 Updating Endpoints

This utility enables you to change an endpoint from active to inactive state or vice-versa.

```
ant -f ant-hcfp-util.xml hcfpupdateendpoint
```

Table 16-7 lists the options for this utility.

Table 16-7   Options for hcfpupdateendpoint utility

Option	Description	Domain	Required
`endpoint`	The name of the endpoint		Yes
`active`	To make the endpoint active	`true` `false`	Yes

Example - **Updates an endpoint from the inactive to active state**

```
ant -f ant-hcfp-util.xml hcfpupdateendpoint -Dendpoint=FileEndpoint -Dactive=true
```

16.9 Pausing and Resuming Endpoints

Use the `hcfppauseendpoint` utility under the ANT `ant-hcfp-util.xml` build file to pause an endpoint.

Table 16-8 lists the options for this utility.

Table 16-8   Options for hcfppauseendpoint utility

Option	Description	Domain	Required
endpoint	The name of the endpoint		Yes

Example - **Pauses an endpoint**

```
ant -f ant-hcfp-util.xml hcfppauseendpoint -Dendpoint=FileEndpoint
```

Use the `hcfpresumeendpoint` utility under the ANT `ant-hcfp-util.xml` build file to resume an endpoint.

Table 16-10 lists the options for this utility.

Table 16-9   Options for hcfpresumeendpoint utility

Option	Description	Domain	Required
endpoint	The name of the endpoint		Yes

Example - Resumes a paused endpoint

```
ant -f ant-hcfp-util.xml hcfpresumeendpoint -Dendpoint=FileEndpoint
```

16.10 Deleting Endpoints

This utility deletes an endpoint.

```
ant -f ant-hcfp-util.xml hcfpdeleteendpoint
```

Table 16-10   Options for hcfpdeleteendpoint utility

Option	Description	Domain	Required
endpoint	The name of the endpoint		Yes

Deletes an endpoint from the inactive to active state

```
ant -f ant-hcfp-util.xml hcfpdeleteendpoint -Dendpoint=FileEndpoint
```

16.11 Updating Keystore

This utility updates the keystore location and password. The updated location and password is visible in Oracle Healthcare console.

```
ant -f ant-hcfp-util.xml hcfpkeystoreupdate
```

Table 16-11   Options for hcfpdeleteendpoint utility

Option	Description	Domain	Required
`keystorelocation`	The location of the keystore		Yes
`keystorepassword`	The password of the keystore		Yes

Example - Updates the keystore location and password

```
ant -f ant-hcfp-util.xml hcfpkeystoreupdate -Dkeystorelocation="/tmp/acme.jks" -
Dkeystorepassword="welcome"
```

For more information about keystores, see "Managing Keystore" in *Oracle Fusion Middleware User's Guide for Oracle B2B*. Note that Healthcare uses the SSHI stripe, so URIs look like this:

```
kss://SSHI/Acme
```

General information about Fusion Middleware security can be found in *Oracle Fusion Middleware Securing Applications with Oracle Platform Security Services*.

A

Back-End Applications Interface

This appendix lists the message properties supported by Oracle SOA Suite for healthcare integration.

The appendix contains the following topics:

- Mapping SCA Normalized Message Properties
- Normalized Message Properties

A.1 Mapping SCA Normalized Message Properties

Table A-1 maps the SCA normalized message properties to the JMS adapter properties.

Table A-1   Healthcare IP_MESSAGE_TYPE to AS11 SCA Normalized Message Property Mapping

SCA	JMS
hc.messageId	MSG_ID
hc.replyToMessageId	INREPLYTO_MSG_ID
hc.fromEndpoint	FROM_ENDPOINT
hc.toEndpoint	TO_ENDPOINT
hc.action	ACTION_NAME
hc.documentTypeName	DOCTYPE_NAME
hc.documentProtocolVersion	DOCTYPE_REVISION
hc.messageType	MSG_TYPE
hc.conversationId	-
hc.interfacesequenceId	INTERFACE_SEQUENCE_ID
hc.interfacesequencediscardid	INTERFACE_SEQUENCE_DISCARD_ID
hc.groupCount	INTERFACE_GROUP_COUNT
-	INTERFACE_GROUP_POSITION

A.2 Normalized Message Properties

Header manipulation and propagation are key business integration messaging requirements. Like other SOA components such as Oracle BPEL Process Manager,

Oracle Mediator, and Oracle JCA, Oracle SOA Suite for healthcare integration relies on header support to solve integration requirements. For example, you can preserve a file name from the source directory to the target directory by propagating it through message headers.

Normalized messages have two parts, properties and payload. Typically, properties are name-value pairs of scalar types. To fit the existing complex headers into properties, properties are flattened into scalar types.

Table A-2 lists the predetermined properties of a normalized message for Oracle SOA Suite for healthcare integration.

Table A-2   Properties for Oracle SOA Suite for healthcare integration

Property Name	Propagable (Yes/No)	Direction (Inbound / Outbound)	Data Type	Range of Values	Description
hc.conversationId	No	Both	String	-	The ID used to relate the message to the message response
hc.documentDefinitionName	No	Both	String	-	The document definition
hc.documentProtocolName	No	Both	String	-	The document protocol
hc.documentProtocolVersion	No	Both	String	-	The document version
hc.documentTypeName	No	Both	String	-	The document type, for example, 850 for an EDI X12 document
hc.fromEndpoint	No	Inbound	String	-	Endpoint name from which an inbound message was received.
hc.messageId	No	Both	String	-	A unique message ID, not directly related to ECID. (ECID information is stored in the B2B AppMessage table.)
hc.messageType	No	Both	String	-	Message type values are: • Request = 1 • Response = 2 • Functional Ack = 9

Table A-2 (Cont.) Properties for Oracle SOA Suite for healthcare integration

Property Name	Propagable (Yes/No)	Direction (Inbound / Outbound)	Data Type	Range of Values	Description
hc.replyToMessageId	No	Both	String	-	The message ID to which the sending message is replying
hc.toEndpoint	No	Outbound	String	-	The endpoint name to which the outbound message is delivered.
hc.interfacesequenceId	No	Both	String	-	For inbound interfaced sequenced messages, this header is delivered to the back end. The composite or application must be designed to read this header and pass the same header to the outbound message. The outbound message must include this header in order to be recognized as an interface sequenced message. This information is used by the sequencing framework to correlate the message.
hc.interfaceGroupCount	No	Outbound	String	-	This header is relevant for the outbound message when the message is fanned out. (A source message is sent to multiple destinations) The composite must populate this header with the number of fan-outs for a given source message when enqueuing the message to the HC adapter.

Table A-2   (Cont.) Properties for Oracle SOA Suite for healthcare integration

Property Name	Propagable (Yes/No)	Direction (Inbound / Outbound)	Data Type	Range of Values	Description
`hc.interfacesequenc ediscardid`	No	Outbound	String	-	Serves as a dummy substitute for an outbound message used to determine whether all messages for the group have arrived.
					For example, a message has to be split into two flows for the lab and pharmacy.
					Based on some criteria in the lab, it is determined the message should not be delivered. To achieve this, the user must send a message to Healthcare with a `GROUP_COUNT` of "2" along with the `INTERFACE_SEQUE NCE_DISCARD_ID` flag.
					The pharmacy flow delivers the message normally with a `GROUP_COUNT` of "2" and the `INTERFACE_SEQUE NCE_ID` flag set.
					Based on the presence of the `INTERFACE_SEQUE NCE_DISCARD_ID` flag, Healthcare determines if the lab message should be processed. The message for count matching is used for the group and then the message is ignored.
					If `INTERFACE_SEQUE NCE_DISCARD_ID` is present alone, it continues to discard the entire message set.

Table A-2   (Cont.) Properties for Oracle SOA Suite for healthcare integration

Property Name	Propagable (Yes/No)	Direction (Inbound / Outbound)	Data Type	Range of Values	Description
INTERFACE_GROUP_POS ITION (JMS Header)	No	Outbound	String	-	Use the INTERFACE_GROUP _POSITION to determine the order of messages when multiple messages within a group (INTERFACE_GROU P_COUNT) are intended for the same endpoint and ordering is enforced among these messages. The message to be delivered first is set with the value "1" and the next message is set with the value "2". The messages are delivered in that order regardless of when these messages are received by the outbound Healthcare adapter.

B

Creating Endpoints with Different Transport Protocols

The appendix covers how to create endpoints with different bidirectional and single-directional transport protocols.

The appendix contains the following topics:

- Creating Bidirectional Endpoints
- Creating Single-Directional Endpoints

B.1 Creating Bidirectional Endpoints

The supported bidirectional protocols are MLLP 1.0, MLLP 2.0, Generic TCP, and HLLP.

B.1.1 Creating an MLLP 1.0 Endpoint

This section covers how to create a bidirectional endpoint with the MLLP 1.0 transport protocol.

To create an endpoint with the MLLP 1.0 transport protocol:

1. Log on to the Oracle SOA Suite for healthcare integration user interface.

2. In the **Configuration** tab under the **Design** tab, click the **Endpoint** folder and then click the **Create** button as shown in Figure B-1.

    Figure B-1   Create Endpoint Button
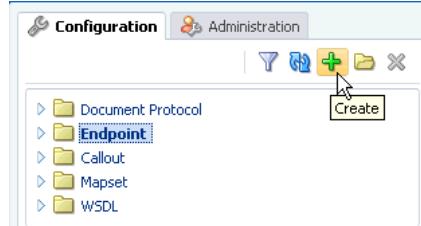
3. In the **Create** window, enter the following and click **OK**, as shown in Figure B-2:

    - **Name**: Name of the endpoint.

    - **Transport Protocol**: Transport protocol for the sending or receiving messages. In this case, select **MLLP10**.

    - **Connection Mode**: Server or Client. If the endpoint is configured as server, Oracle SOA Suite for healthcare integration engine starts listening on a port and

waits for a client to connect to it. In general, the server connection mode is for inbound case. When configured as client, the engine connects to the `hostname` and port of a remote computer or device. In general, this is for an outbound case.

- **Host Name**: In case of an MLLP 1.0 Server endpoint, it should be name or IP address of the computer hosting Oracle SOA Suite, and in the case of an MLLP Client endpoint, it should be the remote host name or device name. Typically, this should be `localhost`. However, Host name can also be the name of the remote host or device.

- **Port**: port number should be more than 500. If the connection mode is set to Server, then the port must be a valid TCP port number. If the connection mode is set to Client, then the port must be the same as the port used on the MLLP server.

This creates the endpoint, and the endpoint is displayed in the right panel of the Oracle SOA Suite for healthcare integration user interface.

Figure B-2   Specifying MLLP 1.0 Endpoint Parameters
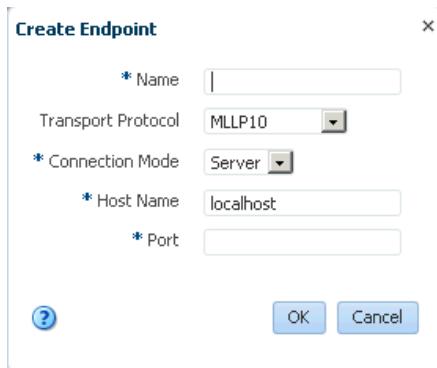
B.1.2 Creating an MLLP 2.0 Endpoint

This section covers how to create a bidirectional endpoint with the MLLP 2.0 transport protocol.

To create an endpoint with the MLLP 2.0 transport protocol:

1. Repeat Steps 1 and 2 from Creating an MLLP 1.0 Endpoint.

2. In the **Create** window, enter the following and click **OK**, as shown in Figure B-3:

- **Name**: Name of the endpoint.

- **Transport Protocol**: Transport protocol for the sending or receiving messages. In this case, select **MLLP20**.

- **Connection Mode**: Server or Client. If the endpoint is configured as server, Oracle SOA Suite for healthcare integration engine starts listening on a port and waits for a client to connect to it. In general, the server connection mode is for inbound case. When configured as client, the engine connects to `hostname` and port of a remote computer or device. In general, this is for an outbound case.

- **Host Name**: In case of an MLLP 2.0 Server endpoint, it should be name or IP address of the computer hosting Oracle SOA Suite, and in the case of an MLLP 2.0 Client endpoint, it should be the remote host name or device name.

Typically, this should be `localhost`. However, Host name can also be the name of the remote host or device.

- **Port**: A valid TCP port number ranging between `1` and `999999`.

This creates the endpoint, and the endpoint is displayed in the right panel of the Oracle SOA Suite for healthcare integration user interface.

Figure B-3   Specifying MLLP 2.0 Endpoint Parameters
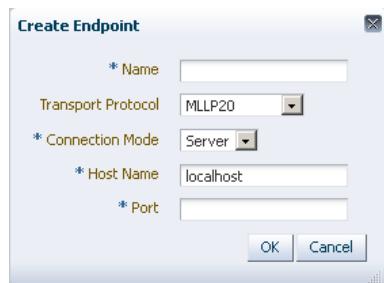
Note:

On the endpoint page, ensure that:

- The Acknowledgement Mode is set to **Sync** for MLLP 2.0. For a Single-Byte Acknowledgment, set the Mode to None.

- In the Transport Protocol Parameters dialog box (displays when you click the **Transport Details** button), the **Permanent Connection** check box is selected in the **Connection** tab.

The reason is that MLLP 2.0 is not supported for Transient connection and for Asynchronous ACK.

B.1.3 Creating a Generic TCP Endpoint

This section covers how to create a bidirectional endpoint with the Generic TCP transport protocol.

To create an endpoint with the Generic TCP transport protocol:

1. Repeat Steps 1 and 2 from Creating an MLLP 1.0 Endpoint.

2. In the **Create** window, enter the following and click **OK**, as shown in Figure B-4:

   - **Name**: Name of the endpoint.

   - **Transport Protocol**: Transport protocol for the sending or receiving messages. In this case, select **GENERIC TCP**.

   - **Connection Mode**: Server or Client. If the endpoint is configured as server, Oracle SOA Suite for healthcare integration engine starts listening on a port and waits for a client to connect to it. In general, the server connection mode is for inbound case. When configured as client, the engine connects to `hostname` and port of a remote computer or device. In general, this is for an outbound case.

   - **Host Name**: In case of an Generic TCP Server endpoint, it should be name or IP address of the computer hosting Oracle SOA Suite, and in the case of an Generic

TCP Client endpoint, it should be the remote host name or device name. Typically, this should be `localhost`. However, Host name can also be the name of the remote host or device.

- **Port**: A valid TCP port number ranging between `1` and `999999`.

This creates the endpoint, and the endpoint is displayed in the right panel of the Oracle SOA Suite for healthcare integration user interface.

Figure B-4 Specifying Generic TCP Endpoint Parameters
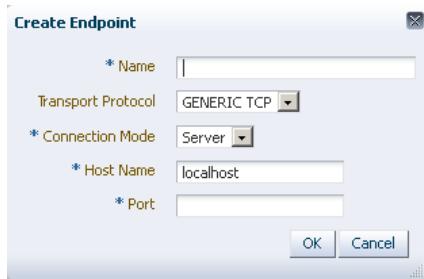
B.1.4 Creating an HLLP Endpoint

HLLP exchange protocol is a variation of the lower layer protocol. It is advanced form of MLLP Exchange Plug-in. It allows error detection and validation of HL7 messages. This protocol is based on TCP transport protocol. This section covers how to create a bidirectional endpoint with the HLLP protocol.

To create an endpoint with the HLLP protocol:

1. Repeat Steps 1 and 2 from Creating an MLLP 1.0 Endpoint.

2. In the **Create** window, enter the following and click **OK**, as shown in Figure B-5:

   - **Name**: Name of the endpoint.

   - **Transport Protocol**: Transport protocol for the sending or receiving messages. In this case, select **HLLP**.

   - **Connection Mode**: Server or Client. If the endpoint is configured as server, Oracle SOA Suite for healthcare integration engine starts listening on a port and waits for a client to connect to it. In general, the server connection mode is for inbound case. When configured as client, the engine connects to `hostname` and port of a remote computer or device. In general, this is for an outbound case.

   - **Host Name**: In case of an HLLP Server endpoint, it should be name or IP address of the computer hosting Oracle SOA Suite, and in the case of an HLLP Client endpoint, it should be the remote host name or device name. Typically, this should be `localhost`. However, Host name can also be the name of the remote host or device.

   - **Port**: A valid HLLP port number ranging between `1024` and `65535`.

   Note:

   You must enable sequencing in the case of HLLP endpoints.

This creates the endpoint, and the endpoint is displayed in the right panel of the Oracle SOA Suite for healthcare integration user interface as shown in Figure B-6.

> **Note:**
>
> In HLLP Client Connection Mode, the **Acknowledgement Mode** dropdown list can only be set to **None**.

Figure B-5   Specifying HLLP Endpoint Parameters
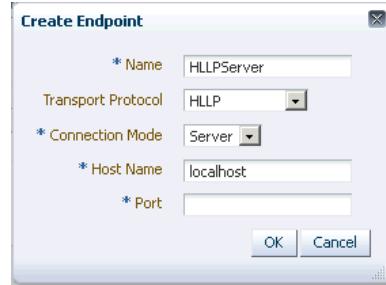
Figure B-6   The Endpoints Page
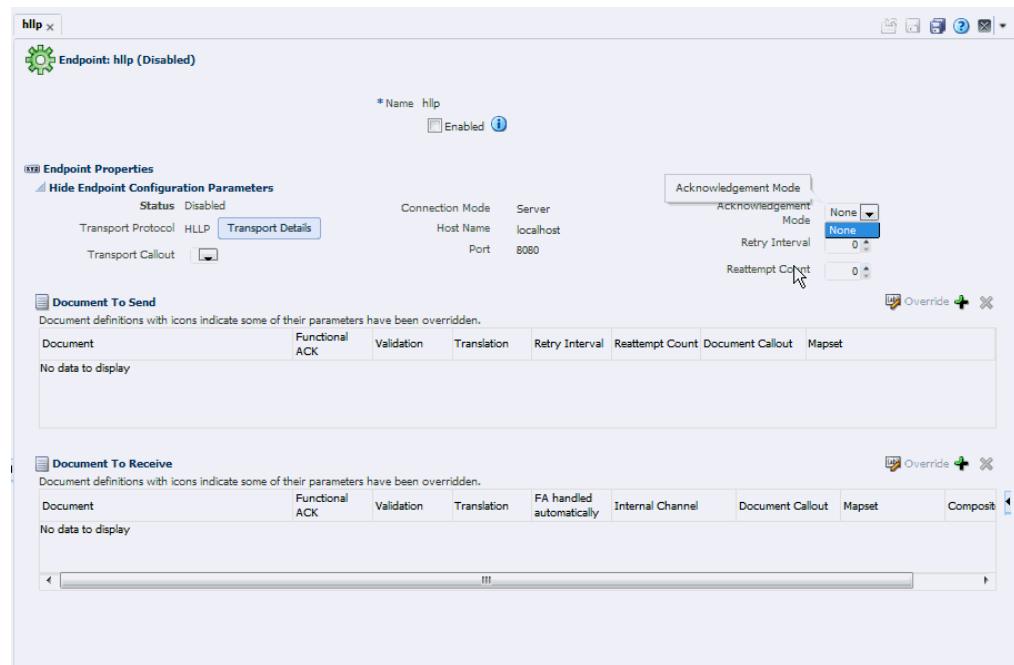
B.2 Creating Single-Directional Endpoints

The supported single-directional protocols are File, FTP, JMS, and SFTP.

B.2.1 Creating a File Endpoint

This section covers how to create a single-directional endpoint with the File transport protocol.

The File transport enables files to be picked up from a shared file directory.

To create an endpoint with the FILE protocol:

1. Repeat Steps 1 and 2 from Creating an MLLP 1.0 Endpoint.

2. In the **Create** window, enter the following and click **OK**, as shown in Figure B-7:

   - **Name**: Name of the endpoint.

   - **Transport Protocol**: Transport protocol for the sending or receiving messages. In this case, select **FILE**.

   - **Direction**: Inbound or Outbound based on your requirement. If the endpoint is configured as inbound, then it can receive response messages or FAs from other endpoints. Conversely, if the endpoint is configured as outbound, it can send messages or FAs.

   - **Folder Name**: An absolute directory path is recommended. Inbound messages are expected in this folder and outbound messages or FAs must be delivered here.

   This creates the endpoint, and the endpoint is displayed in the right panel of the Oracle SOA Suite for healthcare integration user interface.

   Figure B-7   Specifying File Endpoint Parameters
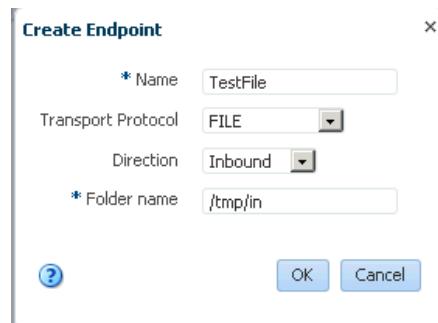
Note:

After a single-directional transport endpoint (inbound/outbound) is created, then it can be edited later to add inbound or outbound configuration by clicking the **Configure** link.

B.2.2 Creating an FTP Endpoint

This section covers how to create a single-directional endpoint with the FTP transport protocol.

FTP enables files to be passed with FTP between applications. FTP runs on default port 21.

To create an endpoint with the FTP protocol:

1. Repeat Steps 1 and 2 from Creating an MLLP 1.0 Endpoint.

2. In the **Create** window, enter the following and click **OK**, as shown in Figure B-8.

   - **Name**: Name of the endpoint.

- **Transport Protocol**: Transport protocol for the sending or receiving messages. In this case, select **FTP**.

- **Direction**: Inbound or Outbound based on your requirement. If the endpoint is configured as inbound, then it can receive response messages or FAs from other endpoints. Conversely, if the endpoint is configured as outbound, it can send messages or FAs.

- **Host name**: The name of the host computer.

- **Folder Name**: An absolute directory path is recommended. Inbound messages are expected in this folder and outbound messages or FAs must be delivered here.

- **User name**: The user name (login name) to connect to the target server.

This creates the endpoint, and the endpoint is displayed in the right panel of the Oracle SOA Suite for healthcare integration user interface.

Figure B-8   Specifying FTP Endpoint Parameters

B.2.3 Creating an JMS Endpoint

This section covers how to create a single-directional endpoint with the JMS transport protocol.

JMS enables applications to send and receive messages to and from the queues and topics administered by any Java Message Service (JMS) provider, including Oracle WebLogic JMS and non-Oracle providers such as MQSeries JMS (IBM). If a user name and password are not provided, the local JNDI is used, including in a clustered environment, provided that the destinations are distributed.

To create an endpoint with the JMS protocol:

1. Repeat Steps 1 and 2 from Creating an MLLP 1.0 Endpoint.

2. In the **Create** window, enter the following and click **OK**, as shown in Figure B-9.

- **Name**: Name of the endpoint.

- **Transport Protocol**: Transport protocol for the sending or receiving messages. In this case, select **JMS**.

- **Direction**: Inbound or Outbound based on your requirement. If the endpoint is configured as inbound, then it can receive response messages or FAs from other endpoints. Conversely, if the endpoint is configured as outbound, it can send messages or FAs.

- **Destination name**: The JNDI name of the JMS queue or topic.

- **Connection factory**: The JNDI name of the connection factory such as `jms/b2b/B2BQueueConnectionFactory`.

This creates the endpoint, and the endpoint is displayed in the right panel of the Oracle SOA Suite for healthcare integration user interface.

Figure B-9   Specifying JMS Endpoint Parameters


B.2.3.1 Retrieving Document Information from JMS Headers

Oracle Healthcare supports retrieving of document information such as DOC_TYPE and DOC_REVISION from JMS headers in the following order:

- If these values are found in JMS headers, then Oracle Healthcare uses these values to identify a document.

- If these values are not found in JMS headers, then Oracle Healthcare uses the payload to identify these values.

> **Note:**
>
> The default `MSG_TYPE` in the case of JMS endpoints are considered as Request. In addition, you can identify the 997 or Acknowledgement messages from `DOC_TYPE`. So, if you want to pass messages with type other than 997, Acknowledgement, or Request, you must explicitly pass `MSG_TYPE` as part of JMS headers.

B.2.4 Creating an SFTP Endpoint

This section covers how to create a single-directional endpoint with the SFTP transport protocol.

SFTP enables files to be passed using SSH FTP. SFTP runs on default port 22, which can be changed to another port.

To create an endpoint with the SFTP protocol:

1. Repeat Steps 1 and 2 from Creating an MLLP 1.0 Endpoint.

2. In the **Create** window, enter the following and click **OK**, as shown in Figure B-10.

   - **Name**: Name of the endpoint.

   - **Transport Protocol**: Transport protocol for the sending or receiving messages. In this case, select **SFTP**.

- **Direction**: Inbound or Outbound based on your requirement. If the endpoint is configured as inbound, then it can receive response messages or FAs from other endpoints. Conversely, if the endpoint is configured as outbound, it can send messages or FAs.

- **Host name**: The name of the host computer.

- **Port number**: A valid SFTP port number within the range of 1 to 999999. The default value is 22.

- **Path**: The absolute directory path where messages are sent from or received.

- **User name**: The user name (login name) to connect to the target server.

This creates the endpoint, and the endpoint is displayed in the right panel of the Oracle SOA Suite for healthcare integration user interface.

Figure B-10   Specifying SFTP Endpoint Parameters

C
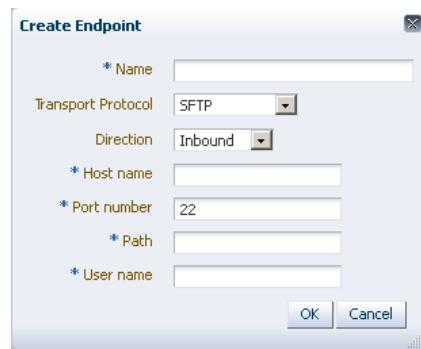
Managing Message Sequencing

This appendix provides information about the command-line and Java-based tools provided by Oracle SOA Suite for healthcare integration to monitor and manage message sequencing. You can use these functions in custom clients.

This appendix includes the following topics:

- Overview of Sequenced Message Management

- Java Methods for Managing Sequenced Messages

- Command-Line Tools for Managing Sequenced Messages

C.1 Overview of Sequenced Message Management

Messages processed through Oracle SOA Suite for healthcare integration are sequenced through the B2B sequence manager. The message sequencing information is stored in the B2B_SEQUENCE_MANAGER database table in order to preserve the correct order of messages. At times, the process of sequencing messages must be managed. For example, if a message in an endpoint becomes stacked and cannot be processed, it can block all other messages for that endpoint from being processed. In this case, manual intervention is required in order to remove or resubmit the message and restart sequencing.

Using the tools provided with Oracle SOA Suite for healthcare integration to monitor and manage message sequencing, you can do the following:

- View all endpoints along with their current state

- View messages by state, endpoint, or a combination of both

- Discard messages by state, endpoint, message ID, or state and endpoint

- Resubmit messages that had errors

- Pause and resume message sequencing for an endpoint.

> **Note:**
>
> You can view the sequenced messages in the Oracle SOA Suite for healthcare integration console. See Working with Sequenced Messages for more details.

C.2 Java Methods for Managing Sequenced Messages

You can use the Java methods provided for sequence message management to create your own clients to monitor and manage the messages in the B2B_SEQUENCE_MANAGER table.

To use the Java methods, create an instance of `SequenceManagerUtility` using the `jndi.properties` file as a parameter. For example:

```
private static Properties getProperties(String s) throws IOException {
    Properties properties;
    FileInputStream fileinputstream = new FileInputStream(s);
    properties = new Properties();
    properties.load(fileinputstream);
    return properties;                 }

SequenceManagerUtility seqUtil = new SequenceManagerUtility(getProperties("/tmp/
jndi.properties"));

seqUtil.listSequenceTargets();
```

For information on creating the `jndi.properties` file, see Prerequisites for Running Command-Line Tools.

C.2.1 Listing Endpoints With States

You can generate a report that lists the messages that are pending in the sequence table with a specific state and for a specific endpoint. The resulting report includes the message IDs and the corresponding state for each.

The syntax of this method is:

```
public java.util.List<java.lang.String> listSequenceTargets()
     throws java.lang.Exception
```

C.2.2 Listing Pending Sequenced Messages

You can generate a report of pending sequenced messages based on the state and endpoint of the messages, based solely on the state, or based solely on the endpoint. This command lists the messages that are pending in the sequence table based on the options you specify. The resulting report includes the message IDs and the corresponding state for each. There are four methods you can use to list sequenced messages, depending on the criteria you want to use.

Listing Pending Sequenced Messages Based on State and Endpoint

The syntax of this method is:

```
public java.util.List<java.lang.String>
getSequenceMessagesByTargetAndState(java.lang.String target, java.lang.String state)
     throws java.lang.Exception
```

Parameter	Description
target	The name (or the IP address and port number) for the endpoint for which you want to list messages. For example: `seqUtil.getSequenceMessagesByTargetAndState("TCP ://111.34.121.22:2025","STACKED")`

Parameter	Description
state	The state of the messages to include in the report. Specify one of the following states: • **PROCESSED**: Message processing is complete and is pending for delivery. • **STACKED**: An error has occurred preventing the message from processing. • **PAUSED**: The endpoint is paused.

Listing Pending Sequenced Messages Based on State

The syntax of this method is:

```
public java.util.List<java.lang.String> getSequenceMessagesByTarget(java.lang.String
state)
      throws java.lang.Exception
```

Parameter	Description
state	The state of the messages to include in the report. Specify one of the following states: • **PROCESSED**: Message processing is complete and is pending for delivery. • **STACKED**: An error has occurred preventing the message from processing. • **PAUSED**: The endpoint is paused.

Listing Pending Sequenced Messages Based on Endpoint

The syntax of this method is:

```
public java.util.List<java.lang.String> getSequenceMessagesByTarget(java.lang.String
target)
      throws java.lang.Exception
```

Parameter	Description
target	The name (or IP address and port number) for the endpoint for which you want to list messages.

C.2.3 Discarding Messages

You can delete sequenced messages from the B2B_SEQUENCE_MANAGER table based on the following criteria combinations:

- State and endpoint of the messages

- State of the messages

- Endpoint of the messages

- Endpoint direction (for endpoint pairs with both inbound and outbound)

- Message ID

- First message only for an endpoint

The discard methods let you manage the messages in the B2B_SEQUENCE_MANAGER table. This is useful in cases where a message is stacked and is blocking other messages from being processed or when there is an issue with a specific message that means it should not be processed sequentially.

Note:

Discarding messages from the sequence manager table does not delete any of the business messages.

Discarding Sequenced Messages Based on State and Endpoint

The syntax of this method is:

```
public java.util.List<java.lang.String>
discardSequencedMessageByStateAnDendpoint(java.lang.String target, java.lang.String
state) throws java.lang.Exception
```

Or:

```
public java.util.List<java.lang.String>
discardSequencedMessageByStateAnDendpoint(java.lang.String target, java.lang.String
state, java.lang.String direction) throws java.lang.Exception
```

Parameter	Description
target	The name (or IP address and port number) for the endpoint associated with the messages to delete.
state	The state of the messages to delete. Specify one of the following states: • **PROCESSED**: Message processing is complete and is pending for delivery. • **STACKED**: An error has occurred preventing the message from processing. • **PAUSED**: The endpoint is paused.
direction	The direction of the messages in the endpoint pair for which you want to discard messages. Specify **INBOUND** or **OUTBOUND**. Only use this option if you are deleting messages for one direction of an endpoint pair.

Discarding Sequenced Messages Based on State

The syntax of this method is:

```
public java.util.List<java.lang.String>
discardSequencedMessageByState(java.lang.String state)
     throws java.lang.Exception
```

Parameter	Description
state	The state of the messages to discard. Specify one of the following states: • **PROCESSED**: Message processing is complete and is pending for delivery. • **STACKED**: An error has occurred preventing the message from processing. • **PAUSED**: The endpoint is paused.

Discarding Sequenced Messages Based on Endpoint

The syntax of this method is:

```
public java.util.List<java.lang.String>
discardSequencedMessageByTarget(java.lang.String target)
     throws java.lang.Exception
```

Or:

```
public java.util.List<java.lang.String>
discardSequencedMessageByTarget(java.lang.String target, java.lang.String direction)
     throws java.lang.Exception
```

Parameter	Description
target	The name (or IP address and port number) of the endpoint for which you want to discard messages.
direction	The direction of the messages in the endpoint pair for which you want to discard messages. Specify **INBOUND** or **OUTBOUND**. Only use this option if you are deleting messages for one direction of an endpoint pair.

Discarding Sequenced Messages Based on Message ID

The syntax of this method is:

```
public java.util.List<java.lang.String>
discardSequencedMessageByMessageId(java.lang.String msgId)
     throws java.lang.Exception
```

Parameter	Description
msgId	The message ID of the message you want to discard.

Discarding the First Message for an Endpoint

The syntax of this method is:

```
public java.util.List<java.lang.String>
discardFirstSequemceMessageByTarget(java.lang.String target)
     throws java.lang.Exception
```

Parameter	Description
target	The name (or IP address and port number) of the endpoint for which you want to discard the first message.

C.2.4 Reprocessing a Message

Reprocessing messages is a useful option to make sure messages are processed in sequence after there is an issue with an endpoint (for example, if the endpoint goes down and then resumes processing).

The syntax of this method is:

```
public boolean processSequenceMessageByMessageId(java.lang.String messageId)
     throws java.lang.Exception
```

Parameter	Description
messageId	The message ID of the message to reprocess.

C.2.5 Pausing and Resuming and Endpoint

There might be times when sequence message processing must be paused for a specific endpoint, such as when an external system fails. In this case, the endpoint can be paused until the system is restored. After the system is restored, you can resume processing for the endpoint. For endpoint pairs, you can pause or resume the endpoint for both inbound and outbound, only for inbound, or only for outbound.

Pausing an Endpoint

The syntax of this method is:

```
public boolean pauseSequenceTarget(java.lang.String target)
     throws java.lang.Exception
```

Or:

```
public boolean pauseSequenceTarget(java.lang.String target, java.lang.String
direction)
     throws java.lang.Exception
```

Parameter	Description
target	The name (or IP address and port number) for the endpoint you want to pause.
direction	The direction of the messages in the endpoint pair you want to pause. Specify **INBOUND** or **OUTBOUND**. Only use this option if you are pausing one direction of an endpoint pair.

Resuming an Endpoint

The syntax of this method is:

```
public boolean resumeSequenceTarget(java.lang.String target)
     throws java.lang.Exception
```

Or:

```
public boolean resumeSequenceTarget(java.lang.String target, java.lang.String
direction)
     throws java.lang.Exception
```

Parameter	Description
target	The name (or IP address and port number) for the endpoint you want to resume
direction	The direction of the messages in the endpoint pair you want to resume. Specify **INBOUND** or **OUTBOUND**. Only use this option if you are resuming one direction of an endpoint pair.

C.3 Command-Line Tools for Managing Sequenced Messages

The command-line tools are run using Apache ant.

These tools are for administrator use only. No security or permission checks are performed to prevent the logged-in user from viewing or discarding data.

C.3.1 Prerequisites for Running Command-Line Tools

Before you can run the command-line tools, you must make sure your environment is configured correctly. Do the following before running any commands:

1. Set the ORACLE_HOME, ANT_HOME, and JAVA_HOME environment variables.

   ORACLE_HOME is your Oracle Fusion Middleware installation directory. For example:

   ```
   set ORACLE_HOME=C:\oracle\Middleware
   set ANT_HOME=%ORACLE_HOME%\modules\org.apache.ant_1.7.1
   set JAVA_HOME=%ORACLE_HOME%\jdk160_18
   ```

2. Create the jndi.properties file.

   ```
   cd $ORACLE_HOME\Oracle_SOA\bin
   ant -f ant-hcfp-util.xml b2bcreate-prop
   ```

3. Edit the jndi.properties file to include the weblogic password.

   Note:

   - After running any command-line tool, restart the healthcare integration user interface or B2B Console. This is because the interfaces cache some metadata and any command-line action that updated the metadata could lead to invalid cached data.

   - All of the command-line tools can be run without any JNDI credentials. To restrict the command-line tools from anonymous use, enter the following information in the jndi.properties file:

     ```
     java.naming.security.principal=weblogic
     java.naming.security.credentials=weblogic_password
     ```

C.3.2 Listing Endpoints With States

You can generate a report that lists all the endpoints and their respective states to help you determine the health of each endpoint.

The syntax of this command is:

```
ant -f ant-hcfp-util.xml hcfpsequencemanager -Dmode=listTargets
```

Option	Description
-Dmode	The mode in which to run the command. For this purpose, set this Opti

C.3.3 Listing Pending Sequenced Messages

You can generate a report of pending sequenced messages based on the state and endpoint of the messages, based solely on the state, or based solely on the endpoint. This command lists the messages that are pending in the sequence table based on the options you specify. The resulting report includes the message IDs and the corresponding state for each.

The syntax of this command is:

```
ant -f ant-hcfp-util.xml hcfpsequencemanager -Dmode=command_mode -
Dstate=message_state -Dendpoint=endpoint_name
```

Option	Description
-Dmode	The mode in which to run the command. For this purpose, set this option to **report**.
-Dstate	The state of the message to include in the report. Specify one of the following states: • **PROCESSED**: Message processing is complete and is pending for delivery. • **STACKED**: An error has occurred preventing the message from processing. • **PAUSED**: The endpoint is paused. To generate a report with all states, do not use this option when running the command.
-Dendpoint	The name (or IP address and port number) of the endpoint for which you want to list messages. To generate a report for all targets, do not use this option when running the command.

Example - Listing Sequenced Messages Based on State and Endpoint

```
ant -f ant-hcfp-util.xml hcfpsequencemanager -Dmode=report -Dstate=STACKED -
Dendpoint=Pharmacy01
```

Example - Listing Sequenced Messages Based on State Only

```
ant -f ant-hcfp-util.xml hcfpsequencemanager -Dmode=report -Dstate=PROCESSED
```

Example - Listing Sequenced Messages Based on Endpoint Only

```
ant -f ant-hcfp-util.xml hcfpsequencemanager -Dmode=report -Dendpoint=Pharmacy01
```

C.3.4 Discarding Messages

You can delete sequenced messages from the SEQUENCE_MANAGER database table based on the following criteria combinations:

* State and endpoint of the messages

* State of the messages

* Endpoint of the messages

* Message ID

* First message only for an endpoint

Discard mode lets you manage the messages in the B2B_SEQUENCE_MANAGER table. This is useful in cases where a message is stacked and is blocking other messages from being processed or when there is an issue with a specific message that means it should not be processed sequentially.

Note:

Discarding messages from the sequence manager table does not delete any of the business messages.

The syntax of this command is:

```
ant -f ant-hcfp-util.xml hcfpsequencemanager -Dmode=command_mode -
Dstate=message_state -Dendpoint=endpoint_name -Dmsgid=message_id
```

Option	Description
-Dmode	The mode in which to run the command. For discarding messages, set this option to **discard**. If you are discarding only the first message in an endpoint, set this option to **discardFirst** and only specify the endpoint.
-Dstate	The state of the messages to discard. Specify one of the following states: • **PROCESSED**: Message processing is complete and is pending for delivery. • **STACKED**: An error has occurred preventing the message from processing. • **PAUSED**: The endpoint is paused. To discard messages of all states, do not use this option when running the command.
-Dendpoint	The name (or IP address and port number) of the endpoint for which you want to list messages. To discard messages for all targets, do not use this option when running the command.
-Dmsgid	The message ID of a specific message to delete. When you use this option, you must only specify the mode and not the target or state.

Example - Discarding Sequenced Messages Based on State and Endpoint

This example deletes all messages from the sequence manager for the Pharmacy01 endpoint with a state of PROCESSED.

```
ant -f ant-hcfp-util.xml hcfpsequencemanager -Dmode=discard -Dendpoint=Pharmacy01 -
Dstate=PROCESSED
```

Example - Discarding Sequenced Messages Based on State

This example deletes all messages from the sequence manager with a state of PROCESSED.

```
ant -f ant-hcfp-util.xml hcfpsequencemanager -Dmode=discard -Dstate=PROCESSED
```

Example - Discarding Sequenced Messages Based on Endpoint

This example deletes all messages from the sequence manager for the Pharmacy01 endpoint.

```
ant -f ant-hcfp-util.xml hcfpsequencemanager -Dmode=discard -Dendpoint=Pharmacy01
```

Example - Discarding the First Sequence Message of an Endpoint

This example deletes the first message for the Pharmacy 01 endpoint from the sequence manager.

```
ant -f ant-hcfp-util.xml hcfpsequencemanager -Dmode=discardFirst -
Dendpoint=Pharmacy01
```

Example - Discarding Sequenced Messages Based on Message ID

This example deletes a single message from the sequence manager, as specified by the message ID.

```
ant -f ant-hcfp-util.xml hcfpsequencemanager -Dmode=discard -
Dmsgid=OAE851ED131B3D6103A00000152F97E9
```

C.3.5 Reprocessing Messages

You can reprocess sequenced messages by using the `hcfpsequencemanager` utility.

The syntax of this command is:

```
ant -f ant-hcfp-util.xml hcfpsequencemanager -Dmode=command_mode -Dmsgid=message_id
```

Option	Description
-Dmode	The mode in which to run the command. For this purpose, set this option to **reprocess**.
-Dmsgid	The message ID of a specific message to reprocess. When you use this option, you must only specify the mode and not the target or state.

Example - Reprocessing a Message

```
ant -f ant-hcfp-util.xml hcfpsequencemanager -Dmode=reprocess -Dmsgid=32797717
```

C.3.6 Pausing and Resuming an Endpoint

There might be times when sequence message processing must be paused for a specific endpoint, such as when an external system fails. In this case, the endpoint can be paused until the system is restored. After the system is restored, you can resume processing for the endpoint.

The syntax of this command is:

```
ant -f ant-hcfp-util.xml hcfpsequencemanager -Dmode=command_mode -
Dendpoint=endpoint_name
```

Option	Description
-Dmode	The mode in which to run the command. For this purpose, set this option to **pause** or **resume**.
-Dendpoint	The name (or IP address and port number) for the endpoint you want to pause.

Example - Pausing an Endpoint

```
ant -f ant-hcfp-util.xml hcfpsequencemanager -Dmode=pause -Dendpoint=Pharmacy01
```

Example - Resuming an Endpoint

```
ant -f ant-hcfp-util.xml hcfpsequencemanager -Dmode=resume -Dendpoint=Pharmacy01
```

D

Interface Sequencing

This appendix discusses how the Oracle SOA Suite for healthcare integration achieves end-to-end First In First Out (FIFO) processing of messages by using Interface Sequencing.

The appendix contains the following sections:

- Introduction

- Configuration Considerations for Interface Sequencing

D.1 Introduction

Oracle SOA Suite for healthcare integration message processing involves the following broad-level steps:

1. The Oracle Healthcare inbound component receives the message from an endpoint and delivers an equivalent XML message to a SOA composite.

2. The composite performs any required transformations, determines the target endpoints for the message, and routes the same/transformed message to the Oracle Healthcare engine.

3. The Oracle Healthcare outbound component then delivers the message to the target endpoints.

Oracle SOA Suite for healthcare integration enables you to perform FIFO processing of messages by using the following approaches:

- Component Sequencing

- Interface Sequencing

Component Sequencing

This approach achieves FIFO by maintaining the sequence of the messages across the boundaries of all the three components: the Oracle Healthcare inbound component, the composite, the Oracle Healthcare outbound component.

Please see https://stbeehive.oracle.com/content/dav/st/Oracle%20SOA %20Suite%20for%20healthcare%20integration/Public%20Documents/ Oracle%20SOA%20for%20Healthcare%20-%20Sequencing%20Guide.docx for more details about Component Sequencing.

Interface Sequencing

This approach achieves FIFO across endpoints, by correlating the messages across the layers and enforcing sequencing at the point of delivery to the target endpoint

For example, when an inbound Oracle Healthcare endpoint receives 20 messages in order, it time-stamps each message on entry to the system and sends them to the

engine for processing in a sequential manner. The engine processes the messages in the sequence and sends the messages to the outbound endpoint for delivery based on the time stamp. This means that the message that arrives at the inbound endpoint is delivered first from the outbound endpoint.

If for some reason, a message with an earlier time stamp gets stuck due to longer time in processing or gets errored out, the other messages with later time stamps do get processed, but are queued in sequence at the outbound endpoint. Those messages do not get delivered out of the outbound endpoint till the time the message with the earlier time stamp (that has got stuck) is processed successfully, and is delivered out of the outbound endpoint. If the message has got errored out, you can try resubmitting the it or you can delete the message from the Sequenced Endpoint Dashboard so that the processed messages waiting in the queue can be delivered.

> **Note:**
>
> In the case of inbound messages, the **Pause**/**Resume** button is disabled for endpoints with end-to-end Interface Sequencing.

This approach reduces the number of sequencing checkpoints, provides a single point of sequence management, and improves performance and scalability of the system.

> **Note:**
>
> In the case of inbound, if the Application message is created successfully, and some failure happens thereafter, the XML payload is persisted for resubmission, which means that the Application message resubmit option is enabled. If the Application message is not created, then you have to resubmit the message from the Wire.
>
> In the case of outbound, if the transformation from XML to native format is successful, then the XML payload is not persisted. However, if the transformation from XML to native has failed, then the XML payload is persisted for further resubmission.

D.2 Configuration Considerations for Interface Sequencing

To enable Interface Sequencing in Oracle SOA Suite for healthcare integration, you must perform some configurations at the endpoint level as well as at the composite level.

D.2.1 Configuring Interface Sequencing at the Endpoint Level

You can configure Interface Sequencing of messages at the endpoint level by selecting the Interface Sequencing check box in the transport protocol configuration.

To configure Interface Sequencing:

1. In the Oracle SOA Suite for healthcare integration console, open the endpoint for which you want to configure Interface Sequencing.

2. Click the **Transport Details** button to display the Transport Protocol Parameters dialog box.

3. Click the **Advanced** tab.

4. Select a **Sequencing Mode**.

5. Select the **Interface Sequencing** check box.

6. Click **OK**, and then click **Apply** on the endpoint page.

Figure D-1 displays the Transport Protocol Parameters dialog box where you configure Interface Sequencing.

Figure D-1   Interface Sequencing
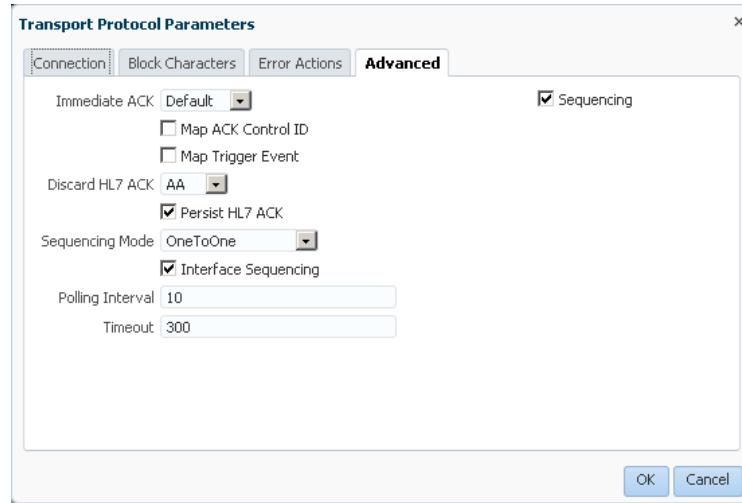
This configuration only affects the inbound messages that are received on an endpoint. The endpoint to which the messages are delivered need not be configured for Interface Sequencing.

After you configure Interface Sequencing at the endpoint level, messages received at this endpoint would be marked for Interface Sequencing, and Interface-Sequencing-specific headers would be delivered to the Internal Delivery Channel and the composite.

Note:

Interface Sequencing does not guarantee the order on how the messages are processed in the composite. This approach only ensures that the messages are delivered to the remote endpoint in a sequential manner.

D.2.2 Configuring Interface Sequencing at the Composite Level

To achieve the correlation (inbound message with outbound) composites handling Interface Sequencing messages are required to process and populate the following additional headers:

- To establish correlation across the transport and engine layers, you must copy the `INTERFACE_SEQUENCE_ID` (or `hc.interfaceSequenceId` in case of fabric) header from the inbound message to the outbound message(s) headers.

- In the case of a fan-out (broadcast), where one inbound message is delivered to multiple endpoints, the composite should populate the

`INTERFACE_GROUP_COUNT` header in each of the outbound message involved in the fan-out.

For example, if a message is to be enqueued to two endpoints, both the enqueued messages should have the header `INTERFACE_GROUP_COUNT` (or `hc.interfaceGroupCount` in case of fabric) value set to 2.

- When fan out messages are intended for the same outbound endpoint, you must use the JMS only property, `INTERFACE_GROUP_POSITION` to maintain the order among the multiple interface outbound messages intended for the same target endpoint. The messages are delivered based on the position value specified. However, in case of in-memory integration on the outbound side, the order in which the message arrives in Oracle Healthcare is used for message delivery.

- In case the composite is required to filter certain messages, a notification must be sent to the sequencing framework to indicate that the message has been skipped.

  This can be achieved by sending a signal to the Oracle Healthcare adapter or the JMS queue with the `INTERFACE_SEQUENCE_DISCARD_ID` (or `hc.interfaceSequenceDiscardId` in case of fabric) header set to the value of `INTERFACE_SEQUENCE_ID` header in the original message.

D.2.3 Understanding Sequenced Message States

Oracle Healthcare provides you the feature to view the states of the interface sequenced messages by using the Dashboard. Figure D-2 displays the sequence state of the message.

Figure D-2   Sequenced Message State
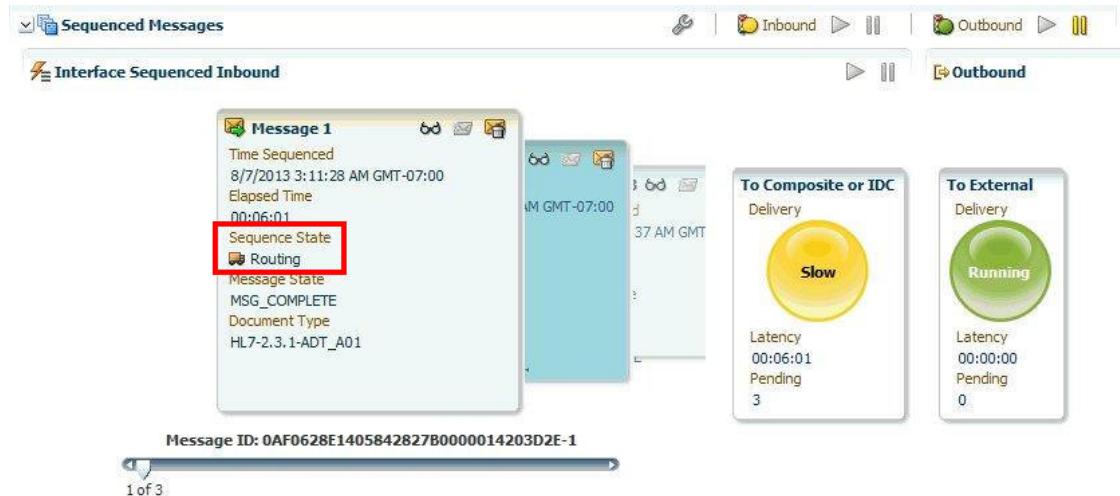
The following sequence states are visible on the Dashboard:

- Routing

- Outbound_Processing

D.2.3.1 Routing

A message with the `Routing` sequence state in the Dashboard indicates that the message has been delivered to the back end. In this case, the row for the inbound message in the `B2B_SEQUENCE_MANAGER` table displays the value as `ROUTING`. The message delivered to the back end has a new header:

- `hc.interfaceSequenceId` in the case of fabric

- `INTERFACE_SEQUENCE_ID` in the case of JMS

D.2.3.2 Outbound_Processing

A message with the `Outbound_Processing` sequence state in the Dashboard indicates that the message has been received from the back end and is currently in outbound processing. In this case, after a message has been delivered to the back end, the message is evaluated if it has the `INTERFACE_SEQUENCE_ID` header, to determine if it is an Interface Sequenced message. Then, if the message has a header, the `INTERFACE_GROUP_COUNT` header is subsequently read to identify a possible fan-out case and the message sequence state is marked as `Outbound_Processing`.

D.2.4 Resubmitting or Discarding Interface Sequenced Messages

You can resubmit or discard interface sequenced messages in case of any error:

- In case of any error in the inbound message processing, you can see that the message has failed in the dashboard. You can then resubmit the message from the Wire.

- In case of any error in the outbound processing, you can just resubmit the Application Message because the payload is persisted.

- You can also discard the outbound message (displayed in the Dashboard) only after all the messages to the target are available. You can also discard the inbound message from the source on the inbound endpoint dashboard. This, in turn, discards any outbound messages to the target for the same sequence. The error message generated contains information of all the discarded messages.

E

Implementing MLLP with High Availability

This appendix describes how to implement Oracle SOA Suite for healthcare integration applications when using the Minimal Lower Layer Protocol (MLLP) in a high availability environment.

This appendix contains the following topics:

- Introduction to Healthcare Integration High Availability
- Enabling MLLP High Availability in Oracle SOA Suite for Healthcare Integration

E.1 Introduction to Healthcare Integration High Availability

High availability for Oracle SOA Suite for healthcare integration is handled through the high availability features of WebLogic Server, Oracle database, and Oracle SOA Suite. You can configure Oracle SOA Suite for healthcare integration for high availability by adding Oracle B2B properties in Oracle Enterprise Manager. These properties enable high availability for healthcare integration projects and specify time out and heartbeat intervals for the servers in the cluster.

All features currently supported for MLLP in Oracle SOA Suite for healthcare integration are also supported in a high availability environment, including message sequencing.

> **Note:**
>
> This SOA Suite feature is part of Oracle Integration Continuous Availability. Please refer to the Oracle Fusion Middleware Licensing Information for more details on Oracle SOA Suite for Middleware Options.

E.1.1 High Availability Processing

In a clustered environment, the first healthcare integration instance to start up and initialize is the instance that handles MLLP traffic. When the instance handling MLLP traffic fails, an inactive instance in the cluster becomes active and takes over the responsibility of handling MLLP traffic within the configured timeout period. All in-flight messages from the failed instance are recovered since message processing is transactional for healthcare integration. This ensures that no messages are lost during failover.

If the instance that fails becomes completely disabled, the second and now active instance continues to pick up messages from and send messages to an outbound distributed queue created specifically for high availability processing. If the initial instance becomes available again, the second instance continues to handle MLLP traffic.

E.1.2 Front-End Failover

A load balancer is used as a failover device in case the active node in the cluster fails. This is required for inbound MLLP traffic when Oracle SOA Suite for healthcare integration is implemented in a clustered environment. The endpoints and external systems use the IP address of the load balancer as the destination connection. This means that all the connections are established in one active node, and message processing is performed across all nodes in the cluster. The load balancer can be configured to distribute the messages evenly among the healthcare integration instances, but only the designated active instance will establish connections. Using cookie-based or active-persistent connections in the load balancer might cause unexpected behavior in the MLLP server. Oracle recommends defaulting to non-persistent connections and verifying the load balancer documentation for persistence settings to eliminate connection losses. Please refer to the load balancer documentation for instructions about how to uniformly farm out the incoming connections across the SOA cluster to the TCP ports (part of Endpoints) configured in the Oracle SOA Suite for heathcare integration user interface.

E.1.3 Notion of Active

The "Active" server is the only server in the cluster that can send or receive messages from an endpoint using MLLP. However, the message processing is done by all servers in the cluster. For example, in the outbound case, messages are prepared to be sent by all the servers in the cluster, but the actual sending is restricted to only the active server. It is the same in the inbound case.

E.1.4 Unit of Order (UOO)

This is an Oracle Weblogic server feature for JMS queues. In the default configuration, all messages belonging to the same UOO are "tied" to a specific server. UOO is used to sequence messages within JMS in the Oracle Weblogic server. So, if UOO is being used, the JMS queue to which the UOO is "tied" has to be active to receive messages. In the event of a server failure, the JMS messages can be recovered using the "Whole Server" migration.

Note:

In an HA environment when external JMS queues are used, a "Whole Server" migration must be configured.

E.1.5 External Dependencies

Oracle SOA Suite for healthcare integration relies on the following components:

- Oracle SOA database for messages and message state persistence

- Metadata Services (MDS) repository for instance metadata

- Load balancer for high availability support

E.1.6 Additional Resources

For more information about configuring Oracle SOA Suite for high availability, see the following:

- "Enterprise Deployment Overview" in *Oracle Fusion Middleware Enterprise Deployment Guide for Oracle SOA Suite*

- Oracle Fusion Middleware High Availability Guide

- "Whole Server Migration" in *Administering Clusters for Oracle WebLogic Server*

E.2 Enabling MLLP High Availability in Oracle SOA Suite for Healthcare Integration

To enable Oracle SOA Suite for healthcare integration, you must define certain B2B properties in Oracle Enterprise Manager. These properties enable high availability for healthcare integration and define time out and ping intervals for the servers.

To enable MLLP high availability for healthcare integration

1. Log on to Oracle Enterprise Manager.

   The URL is http://*hostname*:*port*/em

   where *hostname* is the name of the computer on which WebLogic Server is running and *port* is the port number on which WebLogic Server is listening.

2. In the left navigation panel, expand the **SOA** node and select **soa-infra**.

3. Click the **SOA Infrastructure** menu, point to **SOA Administration** and then select **B2B Server Properties**.

Figure E-1   SOA Infrastructure Menu on Enterprise Manager
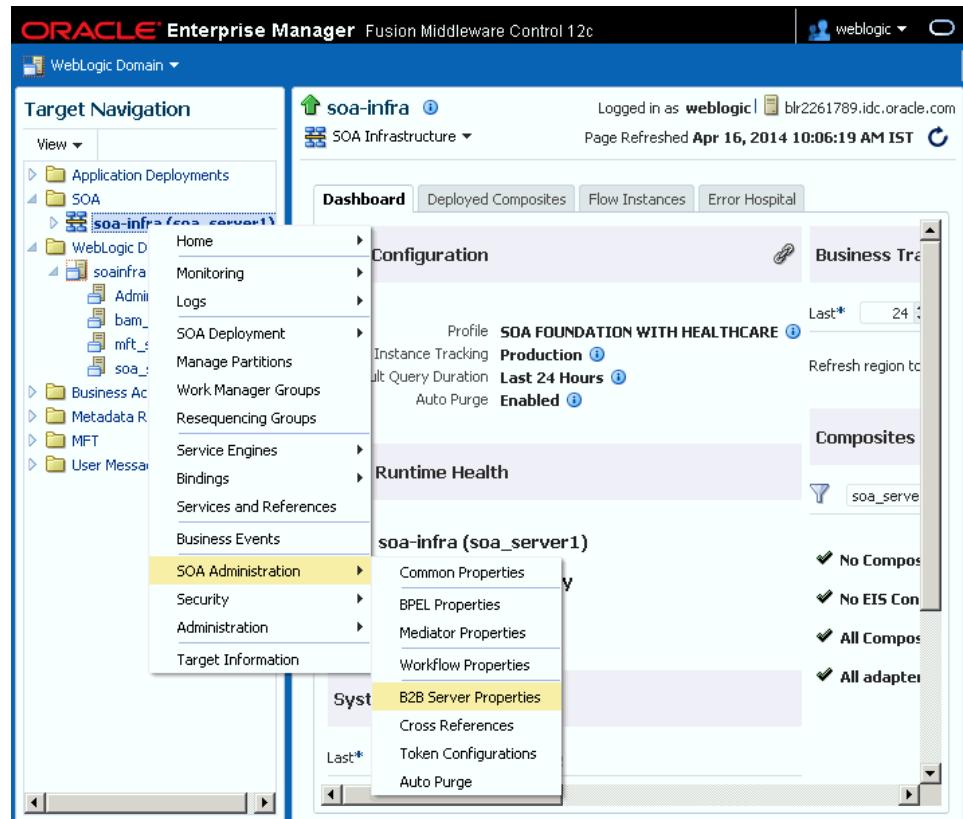
4. On the B2B Server Properties page, click **More B2B Configuration Properties**.

   The System MBean Browser page appears.

5. Click the Operations tab and then click **addProperty**.

Figure E-2   System MBean Browser Operations Page

6. In the Value column of the Parameters table, enter **b2b.HAInstance** in the **key** row and enter **true** in the **value** row.

Figure E-3   Adding the b2b.HAInstance Property
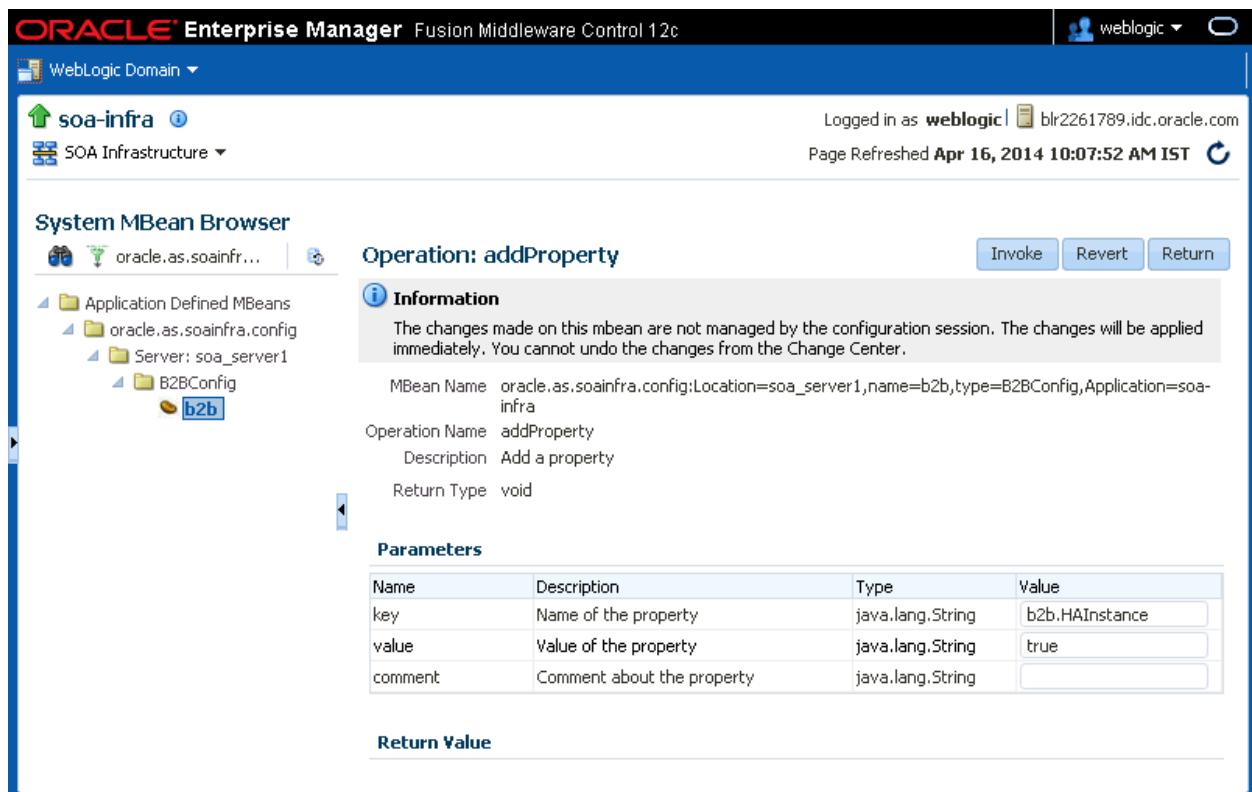
7. Click **Invoke**.

   The new property is saved.

8. Using the previous steps, create a new property named **b2b.MLLP_HA_Mode** and set the value to **true**.

9. You can define the following optional properties. If these are not defined, the default values are used.

Property	Description
b2b.HAMaxElapsedTimeout	The length of time in minutes before the next active instance takes over responsibility for MLLP traffic after the original instance fails. The default value is 2 minutes; 2 is also the minimum value allowed.
b2b.HAHeartBeatPing	The time interval in minutes between pings to determine whether the servers are alive. The default interval is 1 minute; 1 is also the minimum value allowed.
b2b.transportDispatcherThreadCount	The number of MLLP dispatcher threads to use in high availability mode. This is only used for non-sequenced messages, and is used in conjunction with a JMS resource.

Property	Description
b2b.transportDispatcherThreadSleepTime	The length of time (in milliseconds) after which an MLLP dispatcher thread will sleep after message processing.
b2b.MaxTimeinAcquiredState	The length in time in (minutes) that a message is in ACQUIRED state during failover after which it resumes processing.
b2b.AcquiredStatePollingInterval	The length in time in minutes before a message in the ACQUIRED state is polled again.
b2b.SingleTransactionAtInbound	In the case of an inbound MLLP HA, if the server crashes after wire message has been committed to the database but before the event gets enqueued to Event Queue, it is perpetually stuck in the Sequence Manager table and is not processed. This blocks the inbound message flow in the sequencing case.

Set the `b2b.SingleTransactionAtInbound` to true only in the case of MLLP HA to enable the JMS and database commit to take place in a single transaction. It is suitable only for the MLLP case where only one inbound message is received at a time. For example, when messages come in sequentially instead of in parallel for a single endpoint. |

10. When you are done adding properties, click **Return**.

11. After you define high availability properties, you can view them on the Attributes tab. To view the properties, click the Attributes tab and then click **Properties**. Expand the **Element** nodes in the Value table to see the property names and values.

Figure E-4   High Availability Properties in Enterprise Manager

F

Batching HL7 Messages

This appendix provides information on implementing HL7 Batching in Oracle SOA Suite for healthcare integration. Batching allows a batch number of messages to be sent in a single file using an envelope consisting of File and Batch headers. An example would be the Batch of Detailed Financial Transactions (DFTs) sent from an ancillary system (such as laboratory, radiology, pharmacy, and so on) to a financial system.

Using the Healthcare console, you can also specify the number of records to be committed when there is a large number of business messages for a message exchange by using the Partial Batch Commit Size field in the **Administration** tab under **Settings** > **Runtime** > **Miscellaneous**.

This appendix contains the following topics:

- Introduction to HL7 Message Batching

F.1 Introduction to HL7 Message Batching

You can batch HL7 messages based on:

- Batch with File header (FHS)

- Batch with Batch header (BHS)

- Batch with both FHS & BHS

- Batch with only Message header (MSH)

You can batch HL7 messages in two ways:

- Standard Mode: Using HL7 batching protocol

- Custom Mode

F.1.1 Batching with File Header (FHS)

You can configure a default FHS using the Oracle SOA Suite for healthcare integration console.

To configure a default FHS:

1. Log on to the Oracle SOA Suite for healthcare integration console.

2. IN the **Designer** tab, click the **Configuration** tab, and then expand **Document Protocol > HL7> Document Version**.

3. Double-click the Document Version to display the Document Version window on the right-hand pane.

4. Click the **File Header** tab and then select the **Create File Header** check box.

5. Provide the required parameter values in the available fields.

   See Table 3-3 in What You Might Need to Know About HL7 Document Version Parameters for more information on the File Header parameters.

6. Click **Apply**.

Note:

You can create and customize custom File Header `ecs` files by using Oracle Document Editor, and you can use the same to overwrite the default FHS.

F.1.2 Batching with Batch Header (BHS)

You can configure a default BHS using the Oracle SOA Suite for healthcare integration console.

To configure a default BHS:

1. Repeat Steps 1-3 from Batching with File Header (FHS).

2. Click the **Batch Header** tab and then select the **Create Batch Header** check box.

3. Provide the required parameter values in the available fields.

   See Table 3-3 in What You Might Need to Know About HL7 Document Version Parameters for more information on the Batch Header parameters.

4. Click **Apply**.

Note:

You can create and customize custom Batch Header `ecs` files by using Oracle Document Editor, and you can use the same to overwrite the default FHS.

F.1.3 Batching with Message Header (MSH)

In the case of batching MSHs, you can batch all the MSHs separated by a custom delimiter. You can configure this delimiter in the outbound endpoint in the case of Outbound message, and in the listening channel in the case of Inbound message.

See Table 3-3 in What You Might Need to Know About HL7 Document Version Parameters for more information on the Message Header parameters.

F.1.4 Sending Functional Acknowledgments When Batching

When batching messages using the Oracle SOA Suite for healthcare integration console, you can send Functional Acknowledgement to the endpoints partner in the following ways:

- Single file with multiple MSHs

- One Functional Acknowledgement message for every inbound business message (Multiple)

You can specify the Functional Acknowledgement options by using the **Ack Mode** list in the **Miscellaneous** tab in the Document Version window as shown in Figure F-1.

Figure F-1 Specifying Functional Acknowledgment Options
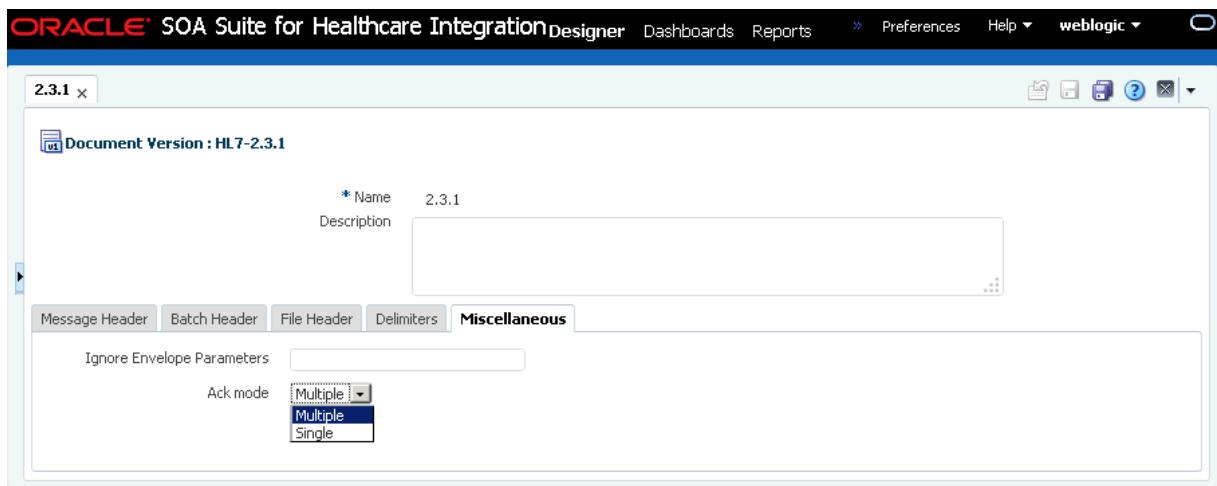
You can also send Acknowledgement based on MSH 15 element. MSH 15 can have values such as:

- AL - always

- ER - Error or rejected conditions only

- NE - Never

- SU - successful completion only

> **Note:**
>
> All batched messages are acknowledged in the response batch. An acknowledgment batch might contain acknowledgment messages only for those messages that have errors.

> **Note:**
>
> - Currently, HL7 batching is supported for a single document over Generic File and Generic FTP protocols.
>
> - Message sequencing and batching do not work simultaneously.

F.1.5 Standard Mode of Batching

You use the HL7 batching protocol in the standard mode of batching.

In this mode, multiple messages are placed in a single file by utilizing the HL7 standard batch protocol (FHS and BHS). A typical HL7 batch file structure looks like the following:

```
[FHS](file header segment)
{--- BATCH begin
```

```
[BHS](batch header segment)
 { [--- MESSAGE  begin
  MSH    (zero or more HL7 messages)
    ....
    ....--- messages may be of same type or different type
    ....
 ] }--- MESSAGE end
[BTS](batch trailer segment)
}--- Batch end
[FTS](file trailer segment)
```

F.1.6 Custom Mode of Batching

In this mode, multiple messages will put into a single file without utilizing the previously mentioned HL7 standard batch protocol (FHS and BHS), instead utilizing some other custom format as shown in this example:

```
MSH
   ....
   ....
## --- message separator
  MSH
    ....
    ....
## --- message separator
  MSH
    ....
    ....
    ....
```

In custom mode, inbound messages are de-batched based on the message separator. The message separator for inbound custom message is derived from the incoming message itself. So no additional configuration is required.

You can specify custom delimiters for de-batching using the Oracle SOA Suite for healthcare integration console.

To specify custom delimiters:

1. Log on to the Oracle SOA Suite for healthcare integration console.

2. Open the required endpoint (File or FTP).

3. Click the **Transport Details** button to display the **Transport Protocol Parameters** dialog box.

4. Click the **Format** tab, specify the custom delimiter in the **Message Splitter** field, and click **OK** to close the dialog box as shown in Figure F-2.

Figure F-2   Specifying Custom Delimiter
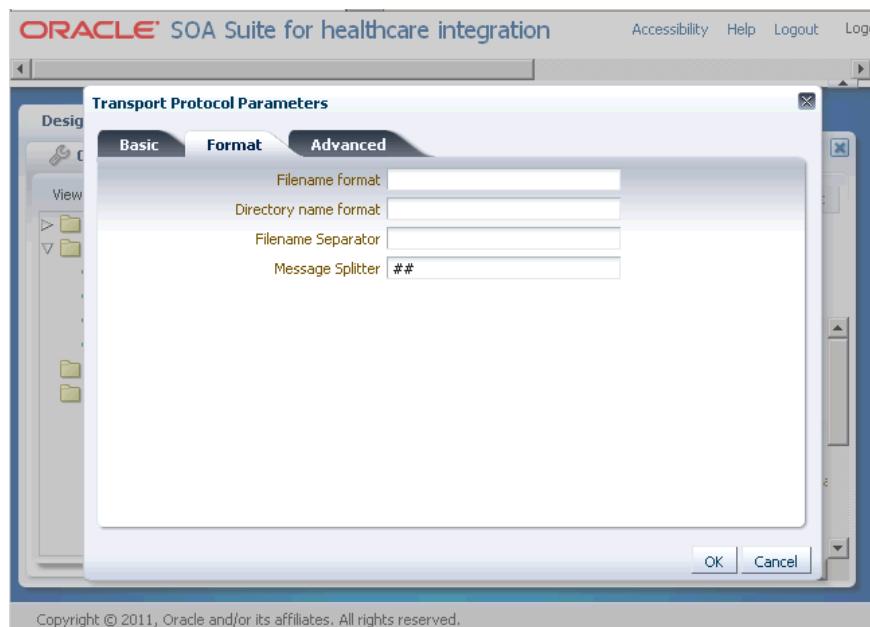
5. Click **Apply** to save the changes made to the endpoint.

> **Note:**
>
> You must ensure that the message separator string should not appear in message payload, otherwise you might see some unexpected errors.

F.1.7 Command-Line Tools for Batching

This sections discusses the command-line tools that are available for batching.

Table F-1 lists the parameters of the batching command line.

Table F-1   Parameters for Batching Command Line

Parameter Name	Description	Domain	Required
endpoint	Name of the endpoint	-	Yes
batchname	Name for the batch	-	Yes
batchtime	The batch trigger time	A cron String or the date in `dd/MM/yyyy HH:mm AM/PM` format	Yes
document	The document protocol name	Values: HL7	Yes
docrevision	The document revision number	-	Yes
doctype	The document type name	-	Yes
isrepetitive	To enable repetitive batching when batch is created using the cron string	`true`|`false` (default)	No

Table F-1   (Cont.) Parameters for Batching Command Line

Parameter Name	Description	Domain	Required
mode	The batch mode	Set to `deletebatch` to delete the batch	No

Examples:

To create a batch operation with FileEndpoint for the ADT_A01 document that is executed in a repetitive mode for the given cron String:

```
ant -f ant-hcfp-util.xml hcfpbatch -Dendpoint=FileEndpoint -Dbatchtime="0 4850 11 7
5 ? 2010"-Dbatchname=batch1234 -Ddocument=HL7 -Ddocrevision=2.3.1 -Ddoctype=ADT_A01 -
Disrepetitive=true
```

To create batches for multiple document types:

```
ant -f ant-hcfp-util.xml hcfbatch -Dendpoint=FileEndpoint -Dbatchtime="0 58 11 7 5 ?
2010"-Dbatchname=batch1234 -Ddocument=HL7 -Ddocrevision=2.3.1 -Ddoctype=ADT_A01
```

or

```
ant -f ant-hcfp-util.xml hcfpbatch -Dendpoint=FileEndpoint -Dbatchtime="07/05/2010
11:45 AM"-Dbatchname=batch1234 -Ddocument=HL7 -Ddocrevision=2.3.1 -Ddoctype=ADYT_A01
```

To delete a batch operation:

```
ant -f ant-hcfp-util.xml hcfpbatch -Dmode=deletebatch -Dbatchname=batch1234
```

If the value for the batchtime contains special characters such as * or # then the character must be escaped using double quotation marks:

```
ant -f ant-hcfp-util.xml hcfpbatch -Dendpoint=FileEndpoint -Dbatchtime='0
5,10,15,20,25,30,35,40,45,50,55,59"" "" "*" ? 2010' -Dbatchname=batch1234 -
Ddocument=HL7 -Ddocrevision=2.3.1 -Ddoctype="ADT_A01" -Disrepetitive=true
```

G

Configuration for Functional Acknowledgment 999

This appendix provides information about the configuration of Oracle SOA Suite for healthcare integration applications to use the document type 999 Functional Acknowledgment (FA).

This appendix has the following topics:

- Introduction
- Function Acknowledgement 999: Use Cases

G.1 Introduction

Oracle Fusion Middleware B2B supports document type 999 Functional Acknowledgment (FA), which is used mostly by HIPAA professionals. By default, the document type configuration for FA is 997. However, you can override the default by specifying document type 999 as the FA. In addition, the document protocol version is not attached to the version of the incoming message, which means that you can send 999 v5010x231 for incoming message 837 v5010x223A1.

G.2 Function Acknowledgement 999: Use Cases

If you use X12 as the document protocol, the default Functional Acknowledgment document type is 997. In case of 997, the version for 997 Acknowledgment is the same as the version of the incoming message. During design time in Oracle B2B console, you can override the default 997 FA or the version, or both in the Document Type Parameters page by using the following parameters:

- Functional Acknowledgment Transaction and
- Functional Acknowledgment Transaction Version

The rest of the configuration, such as Document Definition, Delivery Channel, and Trading Partner Agreement is the same as 997.

The following use cases discusses the configuration for FA 999 and 997:

- Use Case 1
- Use Case 2
- Use Case 3

G.2.1 Use Case 1

If you must use 999 as the FA but want to keep the version the same as the version of the incoming message, then you must set only the Functional Acknowledgment

Transaction to `999` and leave the Functional Acknowledgment Transaction Version blank.

G.2.2 Use Case 2

If you must use 999 as the FA and also want to keep the version different from the version of the incoming message, then you must set both Functional Acknowledgment Transaction and Functional Acknowledgment Transaction Version parameters. For example, you must set Functional Acknowledgment Transaction to `999` and Functional Acknowledgment Transaction Version to `5010X231`.

G.2.3 Use Case 3

If you must use 997 as the FA, but want to keep the version different from the version of the incoming message, you must set Functional Acknowledgment Transaction to `997` and Functional Acknowledgment Transaction Version to `5010`. (assuming that the incoming message version is `4010`). However, this use case is not very common.

Figure G-1 shows the configuration for Functional Acknowledgment Transaction and Functional Acknowledgment Transaction Version.

Figure G-1   Configuration for Functional Acknowledgment Transaction

H

TA1/999 Generation on Error for HIPAA Documents

This appendix discusses the Interchange Acknowledgment (TA1) and document type 999 Functional Acknowledgment (FA) generation on error features of HIPAA documents.

HIPAA messages are used for information exchange. These messages are required to send the TA1 and 999.

The appendix contains the following sections:

- Introduction

- Creating TA1 Documents

- Configuring TA1

- Configuring 999 Acknowledgement on Error

H.1 Introduction

There are many kind of acknowledgments for a HIPAA document that are exchanged between payer and receiver.

When an initiator sends a message, following are the different types of the acknowledgments in the order of precedence:

- TA1 – Interchange level acknowledgment

- 999 – Functional acknowledgment for 5010 versions and higher

The message flow is completed when TA1 or 999 is sent for inbound business message as shown in Figure H-1:

Figure H-1   TA1 Acknowledgment
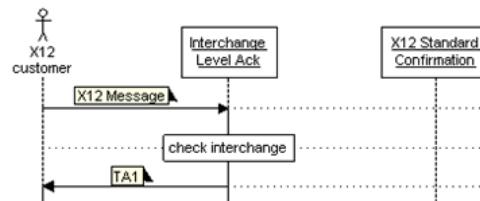
H.2 Creating TA1 Documents

You must create TA1 document type or document definition and trading partner agreement to generate and process TA1 messages. You must create TA1 document type/definition based on Interchange version of X12/HIPAA.

For example, if the Interchange is for "00501" version, TA1 should be created as follows:

1. Open Oracle B2B console by accessing:

   http://*<hostname>*:*<port>*/b2b

   Where *<hostname>* is the name of the computer running Oracle B2B and *<port>* is the port number where Oracle B2B listens, typically `8001`.

2. Click the **Administration** link on top right corner and then click the **Documents** tab.

3. Create Protocol Version `5010` under EDI_X12 interchange version as document protocol version.

4. Create Protocol Document Type `TA1` under `5010` version.

5. Create Protocol Document Definition `TA1Def` (user defined name). When creating the TA1 document definition, TA1.xsd and TA1.ecs are automatically added by Oracle B2B console.

   Figure H-2 displays the creation of a TA1 document.

Figure H-2   TA1 Document Creation
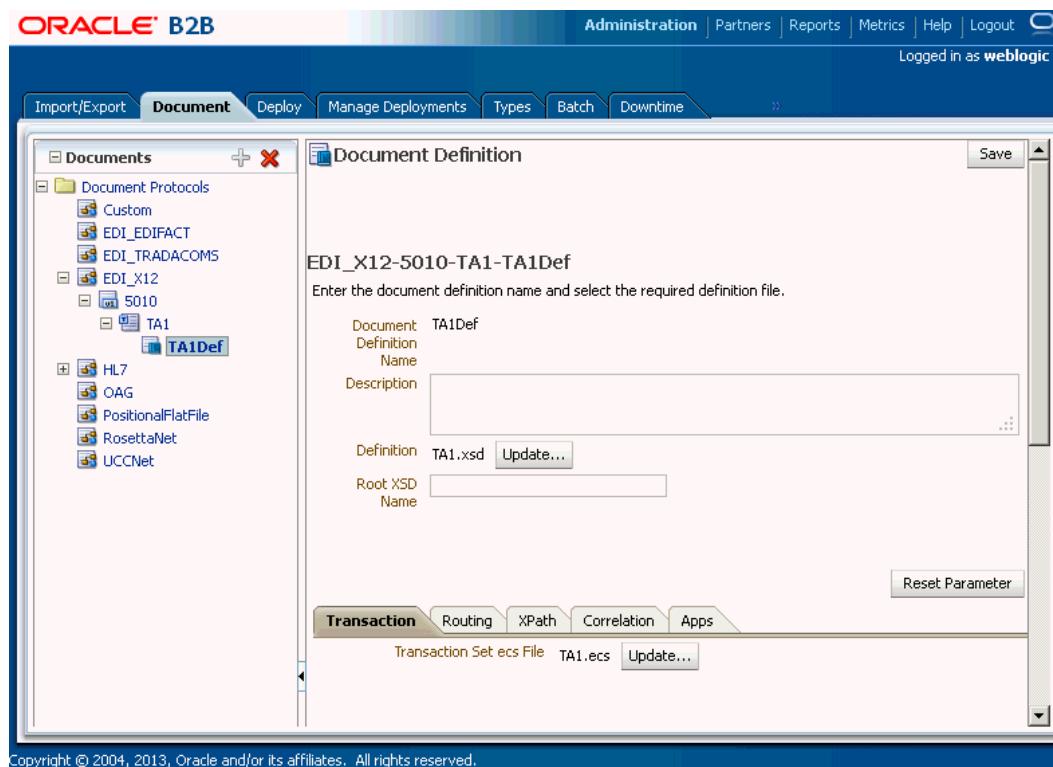
After creating the TA1 document type and definition, you must add the TA1 document definition usage to the required Trading Partner and create the agreement for TA1.

Figure H-3 displays a sample TA1 agreement.

Figure H-3   A Sample TA1 Agreement
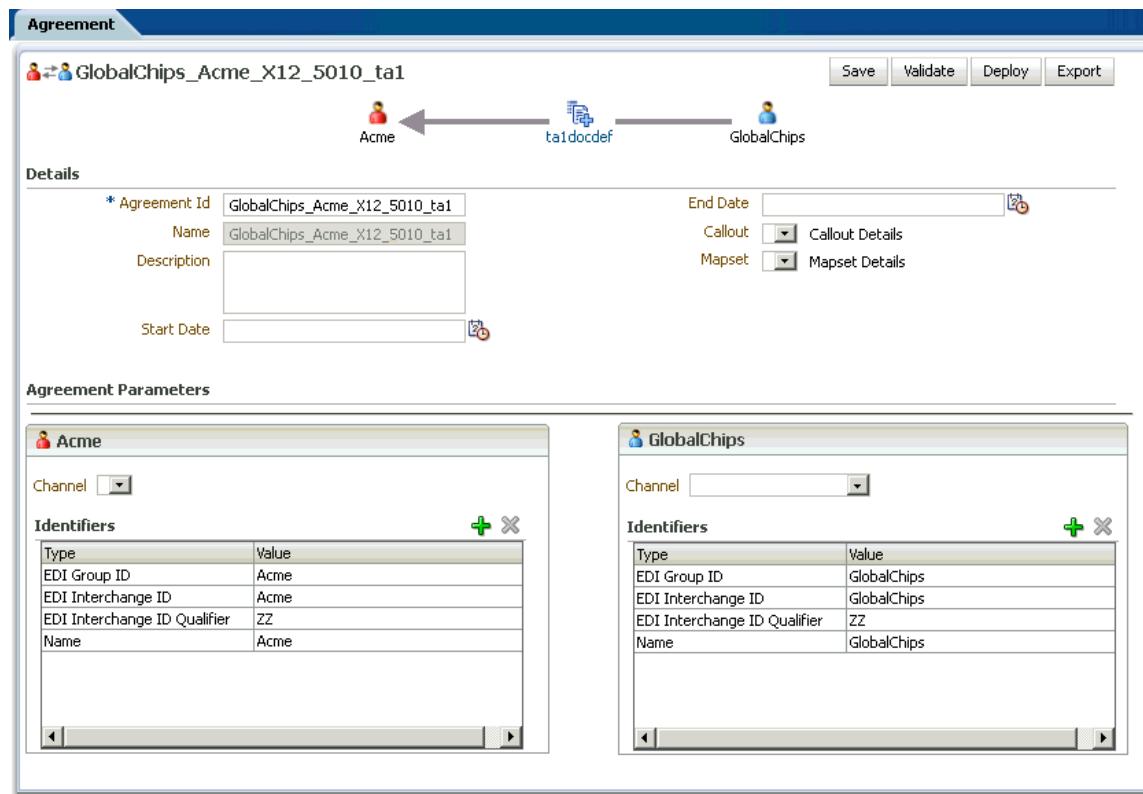
H.3 Configuring TA1

There are two ways to generate TA1 for inbound HIPAA messages. You can either set it at Protocol Version as global or at the Trading Partner level. However, the value set at the Trading Partner level overrides the one set at the Protocol Version level.

The following are the values for TA1:

- Always: TA1 is always generated.

- OnError: A negative TA1 is generated if an invalid Interchange message is received.

- None: No TA1 is generated regardless of valid/invalid Interchange content.

H.3.1 Configuring TA1 at the Protocol Version Level

You can configure TA1 at the Protocol Version level. To configure TA1, open the Protocol Version, such as EDI_X12-5010. In the **Interchange** tab, set the option for TA1 as shown in Figure H-4.

Figure H-4   Configuring TA1 at Protocol Version Level
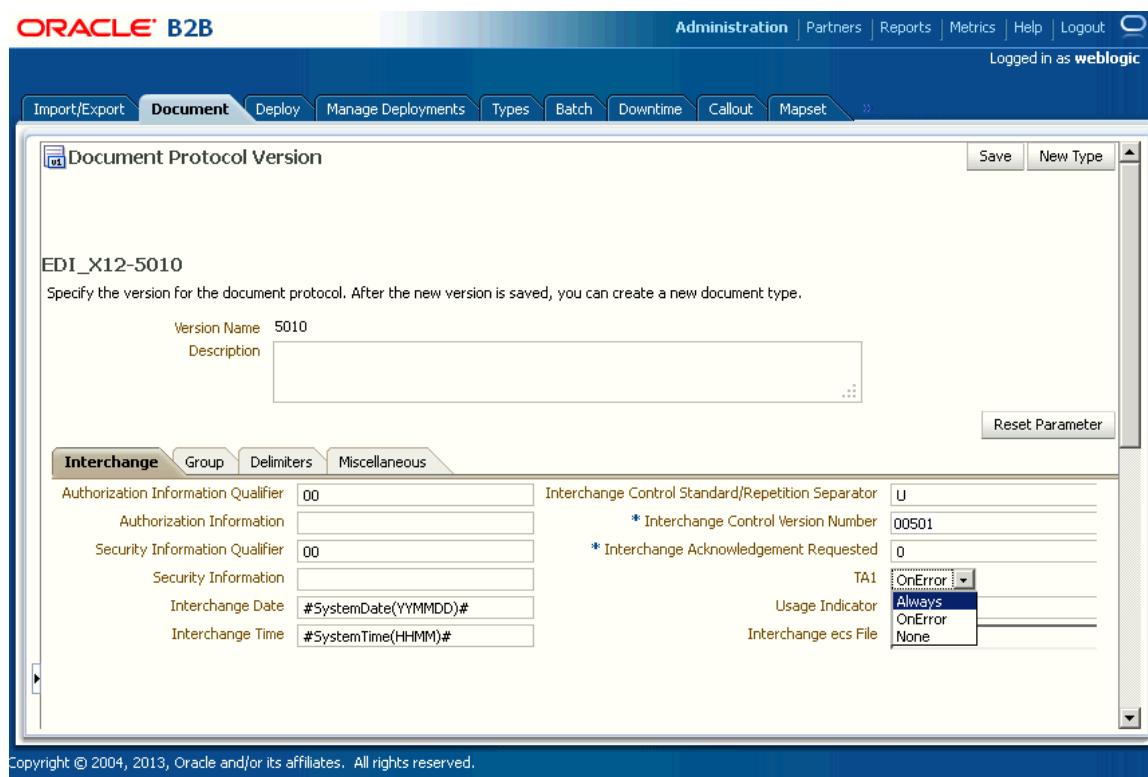
## H.3.2 Configuring TA1 at the Trading Partner Level

You can also configure TA1 at the Trading Partner level. The value set at this level overrides the value set at the global level.

To configure TA1, select the Trading Partner, click the Documents tab, and click the HIPAA document such as EDI_X12-5010-TA1. Under the **Version** tab, click the **Interchange** tab, and set the value of TA1 as shown in Figure H-5.

Figure H-5   Configuring TA1 at the Trading Partner Level
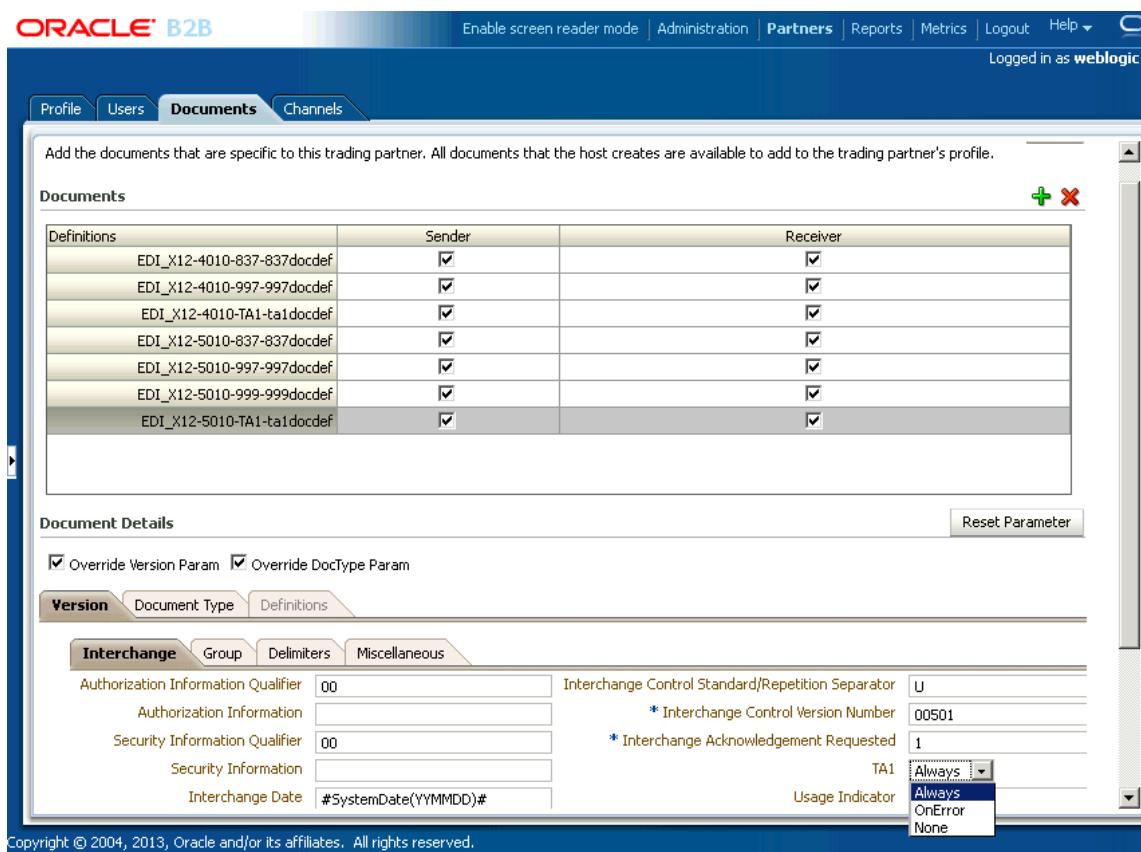
H.3.3 Outbound and Inbound TA1

In the case of an inbound HIPAA message, an outbound TA1 acknowledgement is sent based on the following conditions:

- If the value of the ISA14 segment of a HIPAA message is 0, then no TA1 is generated regardless of valid/invalid Interchange content or if TA1 is set to **Always** or **OnError**.

- If the value of the ISA14 segment of a HIPAA message is 1 and if TA1 is set to **Always**, then a TA1 (positive/negative) is generated.

- If the value of the ISA14 segment of a HIPAA message is 1 and TA1 is set to **OnError**, then a TA1 is generated for invalid Inbound Interchange content.

When an outbound HIPAA message is sent out with ISA14 = 1 (**Interchange Acknowledgement Requested** field in the **Interchange** tab is set to 1 in Figure H-4 or Figure H-5), the outbound message will be in MSG_WAIT_TA1 state until Oracle B2B receives an inbound TA1. However, the MSG_WAIT_TA1 state is overridden when a 999 acknowledgement is received.

H.4 Configuring 999 Acknowledgement on Error

Typically, a 999 Functional acknowledgment is sent when HIPAA messages with version 5010 and higher are received. For this, you must select the Functional Ack check box under the Agreement parameters section in an agreement as shown in Figure H-6.

Figure H-6   Configuring Functional Acknowledgement in the Agreement level
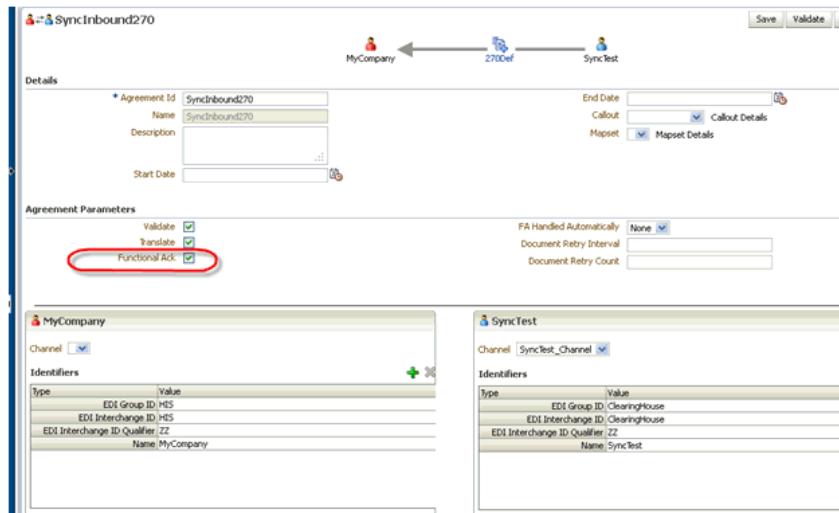
Functional Acknowledgement can be further configured in **Document Type** tab in the Documents page of the Trading Partner to select whether Functional Acknowledgement should be sent only in case of an error as shown in .

Figure H-7   Configuring Functional Acknowledgement on Error Only
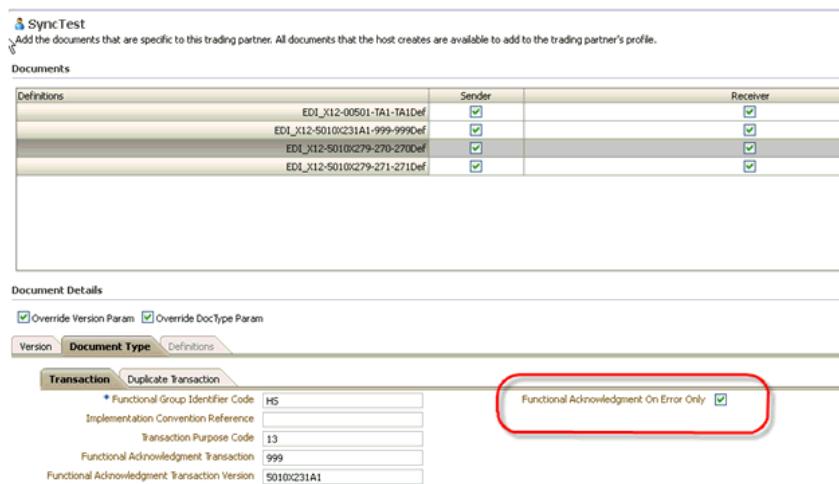
The value of the **Functional Acknowledgement on Error Only** check box is evaluated only if the **Functional Ack** check box is selected at the agreement level. If **Functional Acknowledgement on Error Only** is selected, then Functional Acknowledgement with only reject response is generated for invalid incoming messages; else no Functional Acknowledgement is generated.

I

Implementing SNIP Validation in HIPAA

This appendix discusses how to implement the Strategic National Implementation Process (SNIP) validation for HIPAA messages.

The appendix contains the following sections:

- Introduction

- Configuring SNIP Validations

I.1 Introduction

HIPAA EDI Compliance Check supports data validation and compliance reporting for Type 1 through Type 7. However, from Oracle B2B console, only Type 1-6 are executed.

The types of validation are:

- Type 1 EDI Standard Integrity Testing: Validate basic syntactical integrity of the EDI submission.

- Type 2 HIPAA Implementation Guide Requirement Testing: Validate HIPAA requirement-guide-specific syntax requirement by checking limits on repeat counts, used or not used qualifiers, code, elements, and segments.

- Type 3 HIPAA Balance Testing: Validate that claim line items amounts are equal to total claim amount.

- Type 4 HIPAA Inter-Segment Situation Testing: Validate inter-segment relationship. For example, if element A exists, then element B must be populated.

- Type 5 HIPAA External Code Set Testing: Validate specific code set values for HIPAA standards.

- Type 6 Product Type/Type of Service Testing: Validate that segments that differ based on certain Healthcare services are properly created and processed.

- Type 7 Trading Partner-Specific Testing: Compliance with payer specific requirement. However, this is not implemented in the Oracle Healthcare/B2B set up.

I.2 Configuring SNIP Validations

You can configure SNIP validations at the following levels:

- At the Global level by setting the global severity in XEngine

- At the Document Type level, which is global for all Trading Partners

- At the Trading Partner level

I.2.1 Configuring SNIP at the Global Level

You can set the severity code option in XEngine globally by editing the SeverityConfig.xml located in <SOA_HOME>/soa/thirdparty/edifecs/XEngine/ config directory. You can add the following context to `<SeverityUsage>` section as follows:

```
<ApplyTo>
    <Criteria Name="emp.snip" Value="<snip_type>"/>
    <SetSeverity SeverityID="<severity_type>"/>
</ApplyTo>
```

Where *<snip_type>* takes any value of 1 through 7 indicating the type of SNIP configuration and *<severity_type>* takes the following values for each of the actions that can be specified for each type of validation failure:

- 0 = Ignore

- 2000 = Warning

- 3000 = Information

For example, to validate SNIP levels 1 and 2, you must disable SNIP levels 3-7 by adding the following snippets in the SeverityConfig.xml file:

```
 <SeverityUsage>
...
      <ApplyTo>
          <Criteria Name="emp.snip" Value="3"/>
          <SetSeverity SeverityID="0"/>
      </ApplyTo>
      <ApplyTo>
          <Criteria Name="emp.snip" Value="4"/>
          <SetSeverity SeverityID="0"/>
      </ApplyTo>
      <ApplyTo>
          <Criteria Name="emp.snip" Value="5"/>
          <SetSeverity SeverityID="0"/>
      </ApplyTo>
      <ApplyTo>
          <Criteria Name="emp.snip" Value="6"/>
          <SetSeverity SeverityID="0"/>
      </ApplyTo>
      <ApplyTo>
          <Criteria Name="emp.snip" Value="7"/>
          <SetSeverity SeverityID="0"/>
      </ApplyTo>
...
      </SeverityUsage>
```

After you restart the server, the Oracle Healthcare engine will validate only SNIP levels 1 and 2, and will ignore the rest of the SNIP levels.

For each SNIP validation type, you can specify the following actions:

- **Default**: The Oracle B2B Engine validates the HIPAA message by using the SNIP action configured in XEngine SeverityConfig.xml, which is global for all document types. If there is no option defined in the SeverityConfig.xml, then XEngine will perform a normal validation.

- **Validate**: Indicates that if the data is in error, and it should be fixed prior to further processing. In the case of a production environment, such as XEngine, it indicates that data will not be passed to the next step in the workflow document, and a negative acknowledgement will be sent to the sender of the data.

- **Warning**: Indicates that if problems exist with the data, still the data can continue to be processed. In the case of a production environment, such as XEngine, it indicates that the error will be noted on the acknowledgement document, but the data will pass to the next step in the workflow document. This option will override the option defined in `SeverityConfig.xml`.

- **Information**: Indicates that if the data check reported a message, then the message should be noted. In the case of a production environment, such as XEngine, it indicates that no error will be noted on the acknowledgement document, and the data will pass to the next step in the workflow document.

- **Ignore**: Indicates that any data check message will be suppressed and it will be treated in the same manner as clean data. In the case of a production environment such, as XEngine, it indicates that no error will be noted on the error report or on the acknowledgement document, and the data will pass to the next step in the workflow document.

I.2.2 Configuring SNIP at the Document Level

You can configure SNIP validations at the Document level under the Administration link. Setting the SNIP level at the Document level makes the configuration global for all the Trading Partners that use that particular HIPAA document.

To configure SNIP validation at the Document level:

1. Open Oracle B2B console by accessing:

   http://<*hostname*>:<*port*>/b2b

   Where <*hostname*> is the name of the computer running Oracle B2B and <*port*> is the port number where Oracle B2B listens, typically 7003.

2. Click the **Administration** link on top right corner and then click the **Documents** tab.

3. Open the HIPAA Document Type, such as **EDI_X12-5010-TA1**.Click the **HIPAA** tab.

4. Set the required SNIP validation actions.

Figure I-1 displays a HIPAA document with the various validation types.

Figure I-1   Configuring SNIP at the Document Level
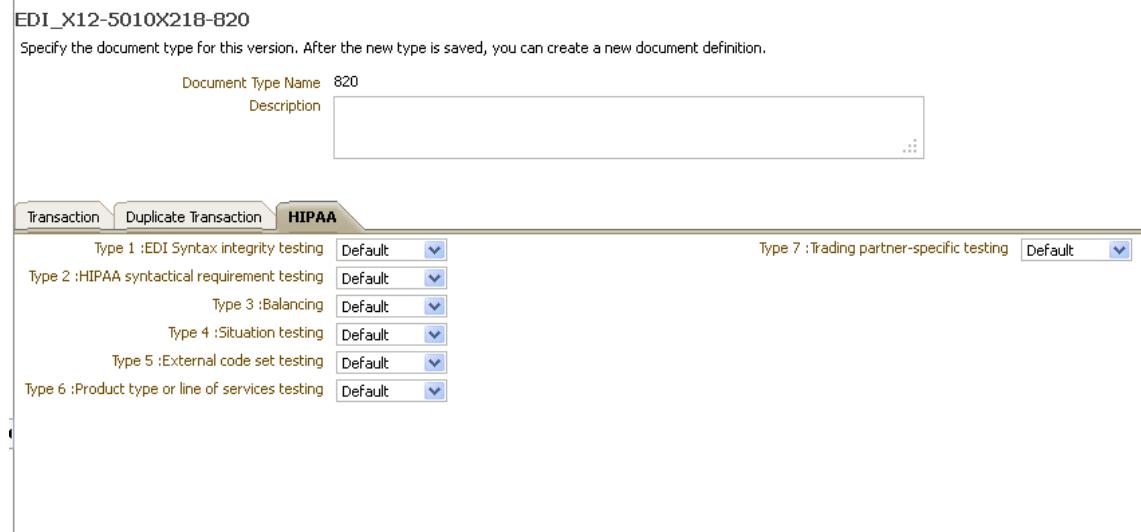
I.2.3 Configuring SNIP at the Trading Partner Level

You can configure SNIP validations at the Trading Partner level under the **Partners** link in the **Documents** tab. Setting the SNIP level at the Trading Partner level overrides the configuration set at the Document Type level.

Figure I-2 displays how to configure SNIP at the Trading Partner level.

Figure I-2   Configuring SNIP at the Trading Partner Level
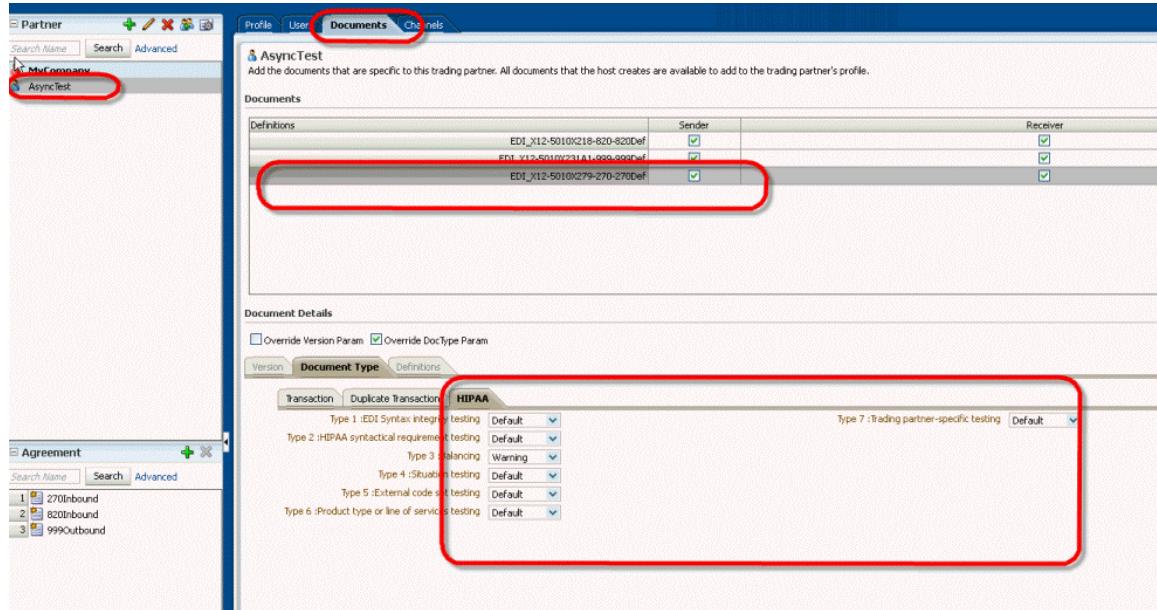
J

Improving Endpoint Scalability by Using NIO

This appendix discusses how linear scalability of a large number of endpoints can be achieved by using the Java New I/O (NIO) API.

This appendix contains the following sections:

- Why Do I Use NIO?

- How Do I Use the NIO Framework?

- Specifying Worker Pool and Selector Pool Size

- Support for MLLP 1.0 Transport Protocol

J.1 Why Do I Use NIO?

Using the old Java IO APIs, issues regarding thread management made it impossible to scale a server to thousands of users. The Java New IO (NIO) framework has been designed to help you to take advantage of the Java NIO APIs in building scalable and robust servers. Using the Java-based socket implementation to manage endpoints with the old IO system causes scalability problems for Oracle SOA Suite for healthcare integration. To overcome this issue, Oracle SOA Suite for healthcare integration uses the NIO-based transport implementation. The NIO framework provides an abstract, event-driven, asynchronous API over various transports such as TCP/IP and UDP/IP.

> **Note:**
>
> NIO is not supported in the cases of Negative ACK and Generic TCP is supported.

J.2 How Do I Use the NIO Framework?

Oracle SOA Suite for healthcare integration uses the NIO support to exchange MLLP messages to:

- Improve latency

- Improve scalability

- Maximize throughput

- Maximize performance

Oracle SOA Suite for healthcare integration uses NIO features seamlessly with the current architecture without affecting existing socket-based IO implementation.To switch to NIO-based transport, you must:

Set the `b2b.nio` server property to `true` in the Oracle Fusion Middleware Enterprise Manager Control console.

> **Note:**
>
> You must restart the server for the property to take effect.

If this property is not set, the default behavior is to use the existing socket-based transport.

> **Note:**
>
> Both transport implementation, which are socket IO and the NIO implementation, cannot work together during runtime. This means that you cannot use the NIO-based transport for one endpoint and the socket-based IO transport for another endpoint.

The NIO-based framework provides the following support for MLLP 1.0 based message exchanges:

- Server and client type socket

- Synchronous and Asynchronous modes of communication

- Transient and permanent connection

- Retry and Timeout policy for message exchange

- Endpoint creation

- Endpoint enabling and disabling

- Endpoint updation

> **Note:**
>
> Using the NIO framework, enabling or disabling 1000 endpoints from the dashboard takes time. It is recommended that you try this feature with a maximum of 100 endpoints at a time.

J.3 Specifying Worker Pool and Selector Pool Size

The NIO framework uses its own thread pool defaults as per the Worker thread pool model. However, Oracle SOA Suite for healthcare integration enables you to specify the Worker thread pool size.

To specify the Worker pool size, set the following properties in the Oracle Fusion Middleware Enterprise Manager Control console:

`b2b.nio.minWorkerPoolSize` = n (default is 3)

`b2b.nio.maxWorkerPoolSize` = *n* (default is 5)

Where *n* is the thread pool size.

Oracle SOA Suite for healthcare integration runtime automatically sets the Selector count value equal to the number of available cores of the CPU of the server. To configure minimum and maximum values of the Selector pool size, set the following properties in the Oracle Fusion Middleware Enterprise Manager Control console:

- `b2b.nio.minSelectorPoolSize`

- `b2b.nio.maxSelectorPoolSize`

> **Note:**
>
> You must restart the server for the property to take effect.

J.4 Support for MLLP 1.0 Transport Protocol

The NIO framework provides support for the MLLP 1.0 transport protocol currently.

It provides support for:

- Support for Immediate acknowledgement

- Support for Discard acknowledgement

- Support for Persist acknowledgement

- Identify TP by Delivery Channel.

- Sequencing Mode:

    - None

    - OnetoOne

    - OnetoOneMapping

- Interface sequencing

- SSL/TLS

K

Audit Reference for Oracle SOA Suite for Healthcare Integration

This appendix provides reference information for auditing in Oracle SOA Suite for Healthcare Integration.

This appendix contains these sections:

- About Custom and Standard Audit Reports
- Audit Events in Oracle SOA Suite for Healthcare Integration

K.1 About Custom and Standard Audit Reports

The Common Audit Framework in Oracle Fusion Middleware provides a set of standard reports based on your audit records. It also enables you to modify the standard reports and create your own custom audit reports.

This appendix provides details about events that can be audited in Oracle SOA Suite for Healthcare Integration. Use this information to understand the structure of each event record to develop custom reports.

For more information, see Oracle Fusion Middleware Application Security Guide.

K.2 Audit Events in Oracle SOA Suite for Healthcare Integration

Table K-1 lists the audit events and their attributes:

Table K-1   Oracle SOA Suite for Healthcare Integration Audit Events

Event Category	Event	Attributes used by Event
UserSession	UserLogin	Date, Time, Initiator, EventType, EventStatus, MessageText, ECID, RID, ContextFields, SessionId, TargetComponentType, EventCategory, ThreadId, FailureCode, RemoteIP, Resource, AuthenticationMethod, Reason
	UserLogout	Date, Time, Initiator, EventType, EventStatus, MessageText, ECID, RID, ContextFields, SessionId, TargetComponentType, EventCategory, ThreadId, FailureCode, RemoteIP, Resource, AuthenticationMethod, Reason

Table K-1 (Cont.) Oracle SOA Suite for Healthcare Integration Audit Events

Event Category	Event	Attributes used by Event
EndpointManagement	EnableEndpoint	Date, Time, Initiator, EventType, EventStatus, MessageText, ECID, RID, ContextFields, SessionId, TargetComponentType, EventCategory, ThreadId, FailureCode, RemoteIP, Resource, AuthenticationMethod, Reason
	DisableEndpoint	Date, Time, Initiator, EventType, EventStatus, MessageText, ECID, RID, ContextFields, SessionId, TargetComponentType, EventCategory, ThreadId, FailureCode, RemoteIP, Resource, AuthenticationMethod, Reason
DocumentManagement	ResubmitMessage	Date, Time, Initiator, EventType, EventStatus, MessageText, ECID, RID, ContextFields, SessionId, TargetComponentType, EventCategory, ThreadId, FailureCode, RemoteIP, Resource, AuthenticationMethod, Reason
	PurgeMessage	Date, Time, Initiator, EventType, EventStatus, MessageText, ECID, RID, ContextFields, SessionId, TargetComponentType, EventCategory, ThreadId, FailureCode, RemoteIP, Resource, AuthenticationMethod, Reason
	ReadPayload	Date, Time, Initiator, EventType, EventStatus, MessageText, ECID, RID, ContextFields, SessionId, TargetComponentType, EventCategory, ThreadId, FailureCode, RemoteIP, Resource, AuthenticationMethod, Reason
Configuration	Import	Date, Time, Initiator, EventType, EventStatus, MessageText, ECID, RID, ContextFields, SessionId, TargetComponentType, EventCategory, ThreadId, FailureCode, RemoteIP, Resource, AuthenticationMethod, Reason
	Export	Date, Time, Initiator, EventType, EventStatus, MessageText, ECID, RID, ContextFields, SessionId, TargetComponentType, EventCategory, ThreadId, FailureCode, RemoteIP, Resource, AuthenticationMethod, Reason

B2B and Healthcare Domain Topology Best Practices

The Oracle Healthcare Adapter entitles healthcare customers to use both the SOA Suite for Healthcare integration solution for HL7 interfaces and the B2B solution for all X12 HIPAA EDI interfaces.

This appendix contains this section:

- Deploy HL7 and X12 HIPAA EDI interfaces in Different Domains

L.1 Deploy HL7 and X12 HIPAA EDI interfaces in Different Domains

From deployment perspective, it is best to install HL7 interfaces (implemented using SOA for Healthcare) and X12 HIPAA EDI interfaces (implemented using the B2B infrastructure) in separate domains for the following reasons:

- **Operation and on-going maintenance**: The healthcare user interface (in SOA for Healthcare) and the B2B console point to the same database instance and both them show HL7 and X12 EDI messages. For that reason, running them in the same domain to show both HL7 and X12 HIPAA messages can create operational, audit, and other maintenance issues.

- **Compliance**: The users of X12 HIPAA EDI data typically should not have access to clinical integration (HL7) data. Displaying both types of data on the screen simultaneously might trouble a risk officer.

- **Performance and scalability**: The SLA for typical X12 HIPAA transactions is different than that required for HL7 documents that include FIFO message sequence processing. Having separate domains for these two transaction sets enhances performance and achieves SLAs.

Additionally, if an enterprise is already using a domain for their B2B transactions, it should *not* upgrade that domain for use as healthcare domain for HL7 transactions too. Instead the enterprise should create an separate domain for HL7 transactions.

The single exception to this policy is the case where the enterprise dumps both X12 HIPAA and HL7 documentation in the database. For example, when adding and/or updating HL7 and X12 HIPAA docs in an EPDR (enterprise patient data repository).

Figure L-1 shows an example SOA healthcare topology that demonstrates the use of separate domains for B2B and Healthcare. The load balancer uses HTTP to route B2B transactions to the B2B domain and uses TCP to route Healthcare transactions to the Healthcare domain.

Figure L-1   Example of SOA Healthcare Topology Using Separate Domains
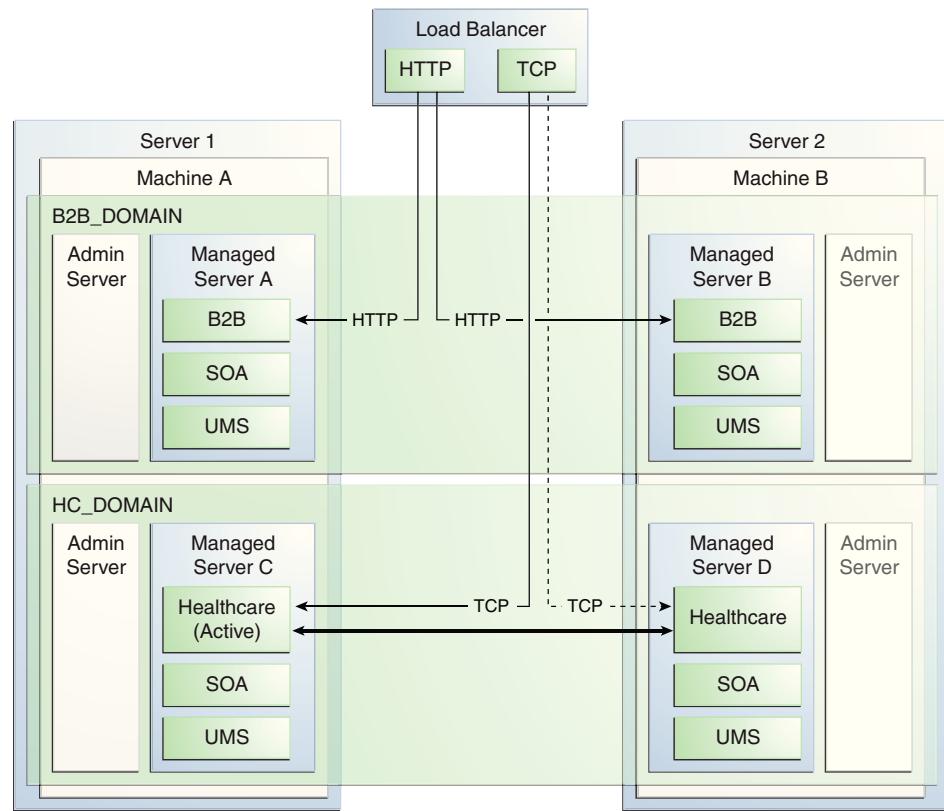
M

Instance Tracking and Error Hospital Integration

This appendix describes instance tracking and error hospital tracking functionality.

This appendix contains the following sections:

- Tracking Messages Between the Oracle Enterprise Manager Fusion Middleware Control Flow Trace and the Healthcare Console

- Tracking the State of a Message from the Oracle Enterprise Manager Fusion Middleware Control Flow Trace XML

M.1 Tracking Messages Between the Oracle Enterprise Manager Fusion Middleware Control Flow Trace and the Healthcare Console

It is possible to track messages between the Oracle Enterprise Manager Fusion Middleware Control flow trace and the B2B/Healthcare console. This functionality is available for both the JMS and in-memory integration modes.

If you intend to track instances and errors across domains for JMS integration then use the following properties:

- `b2b.flowTraceEMURL` (or specify the same at the JMS channel level)

  Use this property to specify the information about the domain that Oracle Enterprise Manager Fusion Middleware Control consumes and uses to send JMS messages from the queue.

- `hc.hcReportsURL`

  Configure this property to point to the SOA Suite for healthcare URL on the instance where the Oracle Enterprise Manager Fusion Middleware Control is running. Also, note that this property is equivalent to the healthcare property of b2b.b2bReportsURL.

These properties are described in more detail in the following chapters of Oracle Fusion Middleware User's Guide for Oracle B2B:

- H Tracking Business Message Flow

- E Setting B2B Configuration Properties in Fusion Middleware Control

M.2 Tracking the State of a Message from the Oracle Enterprise Manager Fusion Middleware Control Flow Trace XML

It is possible to track the complete state of the message from Oracle Enterprise Manager Fusion Middleware Control through the flow trace XML. This functionality is available only for in-memory integration.

In general, the composite instance state follows the application message in Healthcare. As soon as the application message is marked complete, the corresponding composite instance state is updated as complete.

The existing error notification mechanism (sending an exception message to the exception composite/JMS queue) continues to function normally. Whenever possible the notification message is associated with the same flow ID of the original business message.

M.2.1 Inbound Messages

If an inbound document and a composite are deployed to accept the document, there are two types of failure that can be reported.

- **Failure before the document is identified**: The document itself is not identified and composite detection is not possible, therefore no fault reporting occurs.

- **Failure after the document is identified**: The composite is detected and a fault instance is reported in the composite. If an exception composite deployed, the exception composite instance is part of the same flow ID as the reported fault.

M.2.2 Outbound Non-Batch Messages

The following apply to outbound non-batch messages:

- The message remains in the "running" state until the message is delivered to the remote TP.

- Upon successful delivery, the composite instance is marked as being in the "complete" state.

- If an exception occurs during outbound processing, the composite is marked as being in a "faulted" state with the recovery set to B2B_RECOVERY_REQUIRED. The user is expected to recover the message in the B2B/Healthcare console. Note that the recovery state is same for both B2B and healthcare errors.

- The following cases are not handled properly and cause the state to be incorrect in the composite:

  - Negative `Acks` that are received on Healthcare mark the message as in "error". However, this does not update the composite state back to "error".

  - `Ack` time outs are not tracked. The Healthcare message itself is in the "error" state, but the composite remains in the "complete" state.

  - Resubmitting a completed message results in an error. In this case the flow instance state shows up as "complete" despite Healthcare showing it as an error.

    – To track errors in batching, the user must rely on an exception composite to tap into the exception notifications sent to the back-end. The exception message delivered to the back-end is processed as a part of the same flow ID.

M.2.3 Outbound Batch Messages

The following apply to outbound batch messages:

- As soon as the batched message is staged within Healthcare and the message is inserted into the pending message table, the composite instance is marked "complete". Any exceptions that occur beyond this state during the batching process do not affect the state of the original composite.

- The user must rely on an exception composite to tap into the exception notifications sent to the back-end to track batching errors.